MusicScore: A Dataset for Music Score Modeling and Generation

Yuheng Lin
Department of Information Engineering
The Chinese University of Hong Kong
yuhenglin@link.cuhk.edu.hk

Zheqi Dai
Department of Electronic Engineering
The Chinese University of Hong Kong
dzq1367848806@gmail.com

Qiuqiang Kong[*]
Department of Electronic Engineering
The Chinese University of Hong Kong
qqkong@ee.cuhk.edu.hk

Abstract

Music scores are written representations of music and contain rich information about musical components. The visual information on music scores includes notes, rests, staff lines, clefs, dynamics, and articulations. This visual information in music scores contains more semantic information than audio and symbolic representations of music. Previous music score datasets have limited sizes and are mainly designed for optical music recognition (OMR). There is a lack of research on creating a large-scale benchmark dataset for music modeling and generation. In this work, we propose MusicScore, a large-scale music score dataset collected and processed from the International Music Score Library Project (IMSLP). MusicScore consists of image-text pairs, where the image is a page of a music score and the text is the metadata of the music. The metadata of MusicScore is extracted from the general information section of the IMSLP pages. The metadata includes rich information about the composer, instrument, piece style, and genre of the music pieces. MusicScore is curated into small, medium, and large scales of 400, 14k, and 200k image-text pairs with varying diversity, respectively. We build a score generation system based on a UNet diffusion model to generate visually readable music scores conditioned on text descriptions to benchmark the MusicScore dataset for music score generation. MusicScore is released to the public at https://huggingface.co/datasets/ZheqiDAI/MusicScore.

1 Introduction and background

Music scores [1, 2, 3] are written representations of music and contain rich information about music. The visual information in music scores includes notes, rests, staff lines, clefs, dynamics, and articulations. This visual information has advantages over audio [4, 5] and symbolic representations [6, 7] of music in many music modeling and generation tasks, as it contains rich semantic information. Composers create music by writing music scores, making them the most direct modality for representing music.

In this work, we focus on creating MusicScore, a large-scale music score dataset containing image-text pairs for music modeling and generation. The contributions of this dataset are as follows. First, we are the first to propose the *music score generation* dataset and benchmark. We propose that music
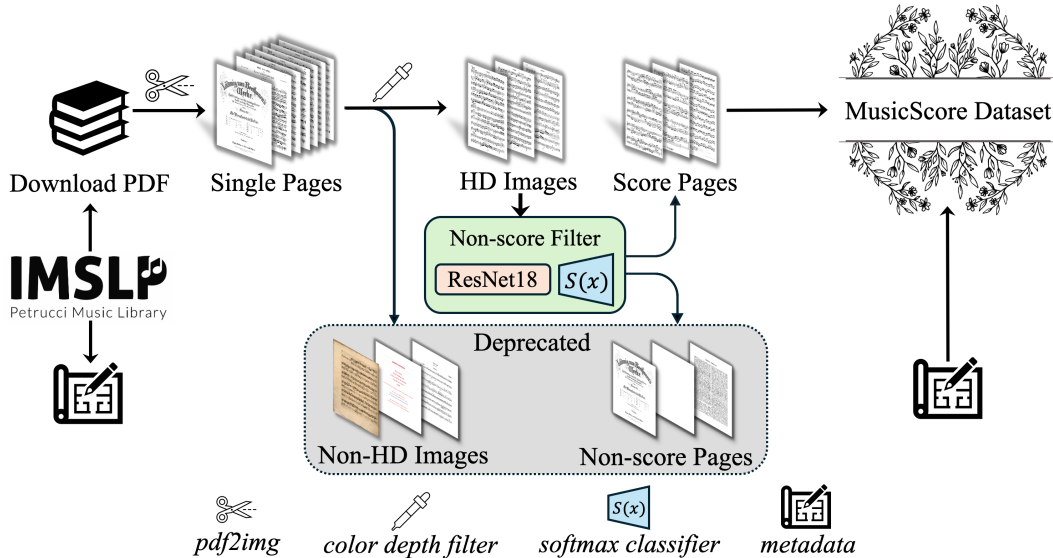
---

[*]Corresponding author.

Figure 1: MusicScore dataset collecting and processing pipeline.

score generation is a novel way of generating music from the vision modality. Second, we collect MusicScore by downloading, processing, and cleaning music scores and their corresponding metadata from the International Music Score Library Project (IMSLP) [8], the largest public music collection in the world. Third, we curate MusicScore into small, medium, and large scales of 400, 14k, and 200k image-text pairs with various diversity, respectively. Fourth, we process MusicScore to contain rich metadata information collected from the general information section of each piece on IMSLP, including but not limited to genre, instrumentation, piece style, etc. Fifth, we build a text-driven latent diffusion system to generate high-quality and playable sheet music images aligned with textual descriptions to benchmark music score generation systems. We release both the MusicScore dataset and the processing code to the public to facilitate research on music score modeling and generation.

2 MusicScore dataset

Different from audio datasets and symbolic music datasets, music scores provide human-readable visual representations. The visual information in scores is important for musicians to analyze and perform music. To fill the gap between the visual modality and music modality for music generation, we collect MusicScore, a large-scale image-text pair dataset for music score modeling and generation. The MusicScore dataset consists of high-quality A4-sized classical music score images. These images are obtained by processing music score PDF files from IMSLP [8]. The metadata for each music piece is obtained by collecting the corresponding general information section from the IMSLP page. MusicScore is carefully filtered and cleaned to ensure the quality of the metadata. We describe the collection and processing of MusicScore as follows.

2.1 Data source

The International Music Score Library Project (IMSLP) [8], commonly referred to as the Petrucci Music Library, is an esteemed subscription-based digital library that houses a vast collection of public-domain music scores and is renowned as the world's largest electronic sheet music archive. IMSLP serves as a valuable resource for musicians and scholars worldwide. By providing unrestricted access to an extensive repertoire of musical compositions, IMSLP promotes the dissemination and preservation of musical knowledge and heritage [9].

Figure 1 shows the music collection and processing pipeline. We download music scores from the IMSLP website [8] in PDF format. Each PDF file may contain multiple pages. In May 2024, we downloaded a total of 148,257 PDF music scores.

2.2 Image processing

Among those 148,257 PDF music scores, not all of them are of high-quality for music score generation. The music scores in IMSLP vary in quality. First, many handwritten scores are manuscripts written by composers who lived centuries ago. Second, many PDF files contain catalogs, textual descriptions, and other front pages that are not relevant to music scores. Third, there are also a number of scores that have been manually scanned and uploaded, containing page misalignment errors or unclear symbols due to low scanner resolution. Figure 4 shows examples of two low-quality and two high-quality scores, respectively. We designed the following criteria to clean the music scores.

- **High Quality** High-quality images are usually created by publishers using professional tools or software such as MuseScore [10], Sibelius [11], and Finale [12]. The color of high-quality scores should be in grayscale with three RGB channels.
- **Full Page Score** In each page of a music score PDF, the proportion of musical content should occupy more than two-thirds of the page. A page should not contain too many texts irrelevant to the music content, such as those on the front cover page.

In order to create a high-quality single-page music score dataset that satisfies the above criteria, we applied three dataset processing stages as follows.

Filter dataset by color depth We extract the color depth information of the PDF files. The color depth of 1-bit corresponds to black and white images, while the color depth of 8-bit or 16-bit corresponds to color images. Some existing works apply the binarization operation to convert color scores into black and white scores. However, the binarization operation often results in blurry or noisy black and white scores, with parts of notes and staves missing after binarization. To ensure the high quality of our dataset, we only retain the music scores with a color depth of 1-bit and remove the music scores with color depths larger than 1-bit. We implement this using the PyMuPDF[2] Python package. After filtering, there are 32,307 PDF files remained in the dataset.

Filter dataset by removing non-score pages We partition all PDF files into separate pages by using the *pdf2img*[3] Python package. We name each page of a music score as *scoreID_pageIndex.jpg*, where *scoreID* is the unique score ID and *pageIndex* is the page index number, respectively. For example, the name *IMSLP514102_17.jpg* corresponds to the 17th page of the music piece *IMSLP514102*.

We train a classification model to classify score and non-score images. This model helps us exclude images that contain insufficient music content, such as cover pages and pages with textual descriptions, from the dataset. Through this step, we ensure that the selected single-page sheet music images only contain music contents. Our model achieves an accuracy of 98% on the test set. After removing the non-score pages, there are 200,480 images in JPG format remained. We refer to this dataset as MusicScore-200k. We then randomly sample 14,656 images from MusicScore-200k to create a medium-scale subset, MusicScore-14k, for the rapid development of music score generation systems.

Manually labelled subset Training with large-scale datasets such as MusicScore-14k and -200k can be computationally intensive, not all users have access to sufficient computational resources. In order to lower the barrier for using MusicScore, we manually curated a small-scale dataset called MusicScore-400, consisting of 403 image-text pairs. The MusicScore-400 subset is not part of the MusicScore-200k. Instead, MusicScore-400 contains music scores partitioned from 19 compositions in history, composed by renowned musicians such as Bach, Beethoven, and Chopin. By providing this smaller dataset, we aim to make it more accessible for researchers with limited resources to explore and experiment with the music score image data, encouraging a wider range of engagement in tasks related to music score images and music generation.

2.3 Metadata processing

We collect metadata from the general information section of each composition's IMSLP webpage. The metadata is stored in a JSON file using the data structure of a Python dictionary and is associated with the PDF files through name IDs. The metadata contains rich information such as the work title,

---

[2] https://pypi.org/project/PyMuPDF/
[3] https://pypi.org/project/pdf2image/

Work Title	Muisca - El Dorado
Alternative Title	A cappella motet
Composer	Steer, Michael Maxwell
I-Catalogue Number	None [force assignment]
Key	G minor
Year/Date of Composition	2018
First Publication	2021
Librettist	Michael Maxwell Steer (b. 1946)
Language	English
Average Duration	3'30" minutes
Composer Time Period	Modern
Piece Style	Modern
Instrumentation	mixed chorus (SATB) a cappella
ID	IMSLP690834

Figure 2: **A sample of MusicScore.** Left: metadata of "Muisca - El Dorado" by Michael Maxwell Steer. Right: The first page score.

composer, key, piece style, instrumentation, etc. Figure 2 shows an example of metadata and the corresponding cover page of a score.

3 Experiments

We propose a novel task called music score generation along with its benchmark system as follows. As a result, we achieved a latent diffusion model capable of generating readable music score images conform to input text description.

3.1 Music score generation system

The system founds on a text-driven latent diffusion model [13], *i.e.*, Stable Diffusion 2.0, as a generative engine which consists of a variational autoencoder [14], a text encoder and a UNet [15] backbone.

Training We fine-tuned a Stable Diffusion model [13] based on the stable-diffusion-2-base pretrained weight (`512-base-ema.ckpt`,[4] SD2.0-base for short) by Stability AI. SD2.0-base is trained from scratch 550k steps at resolution 256×256 on a subset of LAION-5B [16], and is further trained for 850k steps at resolution 512×512 on the same dataset on images with a resolution larger than 512×512. The training consists of two phases. For the first phase, the VAE is fine-tuned using a dataset of 240 sheet piano and violin score images at a resolution of 512×512. We optimize VAE's parameters using Adam [17] with a learning rate of 1×10^{-5}, a linear learning rate scheduler, no weight decay, and a batch size of 8.

The second phase involves fine-tuning the UNet-based diffusion model. We utilize the MusicScore-400 subset to train the diffusion model while keeping the weights of the fine-tuned VAE frozen. The text encoder used is OpenCLIP-ViT/H [18, 19, 20]. We select five categories from the metadata, including composer, instrumentation, piece style, key, and genre, to form the text condition. An example text condition is *a music score, composer is [composer], instrumentation is [instrumentation], key is [key], piece style is [piece_style], genre is [genre]*, where the values in the square brackets are arguments configured by users. By training on MusicScore-400 subset at 512×512 resolution for 78,000 iterations, we achieve a music score generation system which is capable of generating playable music score image via text input. We conduct training using 8 RTX 4090 GPUs with a global batch size of 64. The gradient optimizer is AdamW [21] without parameters modifications from PyTorch's [22] implementation. The learning rate is 1×10^{-5}, no learning rate scheduler is applied.

---

[4] https://huggingface.co/stabilityai/stable-diffusion-2-base

(a) a music score, instrumentation is violin, key is A major

(b) a music score, instrumentation is violin

(c) a music score, instrumentation is piano, key is A major

(d) a music score, instrumentation is piano

Figure 3: Generated music scores. The selected samples are generated by the text-driven Stable Diffusion model fine-tuned on MusicScore-400 at 512×512. For all results, the *classifier-free guidance* scale remains 4.0 and 250 DDIM sampling steps. The input text prompt follows the pattern of: *a music score, instrumentation is [instrumentation], key is [key].*

Evaluation We measure the performance of music score generation using Fréchet Inception Distance (FID) [23], which is a standard metric for evaluating generative models of images. In Table 1, FID-*n* indicates we randomly select n images for each evaluation, where $n \in [8, 16, 32, 64]$. The SD2.0-base is fine-tuned on 512×512 resolution MusicScore-400 for 78,000 iterations. We evaluate FID [24] across MusicScore-400, MusicScore-14k, and MusicScore-200k datasets, the ground-truth images from each subsets are resized to same resolution with generated images at 512×512. During generation, we apply a DDIM sampler [25] for 250 DDIM sampling steps. The text prompt formats remain consistent between training and generation phases. We apply *classifier-free guidance* [26] with guidance strength $\omega = 4.0$ in generation. Table 1 presents the music generation results. We achieve FID scores of 74.46, 229.16, and 261.28 when evaluated on 64 images from the MusicScore-400, MusicScore-14k, and MusicScore-200k datasets, respectively. Figure 8 shows the generated music scores conditioned on text prompts. The first column displays a correctly generated A-major violin score. The third and fourth columns show correctly generated A-major piano scores, indicated by the number and pattern of sharp symbols ♯.

Table 1: **Benchmarking text-conditional Stable Diffusion trained on MusicScore-400 with resolution** 512×512. FID-*n* represents the number of evaluated images from respective subset of MusicScore.

Subset	MusicScore-400	MusicScore-14k	MusicScore-200k
FID-8	114.65	297.60	294.76
FID-16	85.81	221.42	314.06
FID-32	84.33	255.00	264.02
FID-64	74.46	229.16	261.28

4 Conclusion

In this work, we introduce MusicScore, a large-scale music score dataset collected and processed from the International Music Score Library Project (IMSLP). MusicScore consists of image-text pairs, where the image represents a page of music score and the text contains metadata about the music. We provide details about the dataset collection and cleaning processes. MusicScore includes metadata extracted from the general information section of IMSLP pages, which covers composer, instrumentation, era, and other general information of the composition. MusicScore is curated into small, medium, and large scales, with 400, 14k, and 200k image-text pairs, respectively, offering various levels of diversity. MusicScore is publicly released. Additionally, we develop a music score generation system based on UNet diffusion models to generate music scores conditioned on text descriptions, aiming to benchmark MusicScore for music score generation tasks. In the future, we plan to develop a MusicScore-CLIP model for music score modeling. We also intend to integrate score, audio, and symbolic representations to create unified music modeling and generation systems.

References

[1] Jim Jones, Diego de Siqueira Braga, Kleber Tertuliano, and Tomi Kauppinen. Musicowl: The music score ontology. In *Proceedings of the International Conference on Web Intelligence*, pages 1222–1229, 2017.

[2] Nicholas Cook. *Beyond the score: Music as performance*. Oxford University Press, USA, 2013.

[3] Roger B Dannenberg and Christopher Raphael. Music score alignment and computer accompaniment. *Communications of the ACM*, 49(8):38–43, 2006.

[4] Jean-Pierre Briot, Gaëtan Hadjeres, and François-David Pachet. Deep learning techniques for music generation–a survey. *arXiv preprint arXiv:1709.01620*, 2017.

[5] Sander Dieleman, Aaron Van Den Oord, and Karen Simonyan. The challenge of realistic music generation: modelling raw audio at scale. *Advances in Neural Information Processing Systems*, 31, 2018.

[6] Shulei Ji, Xinyu Yang, and Jing Luo. A survey on deep learning for symbolic music generation: Representations, algorithms, evaluations, and challenges. *ACM Computing Surveys*, 56(1):1–39, 2023.

[7] Mingliang Zeng, Xu Tan, Rui Wang, Zeqian Ju, Tao Qin, and Tie-Yan Liu. Musicbert: Symbolic music understanding with large-scale pre-training. *arXiv preprint arXiv:2106.05630*, 2021.

[8] Project Petrucci LLC. International music score library project. `https://imslp.org/`.

[9] Wikipedia. International music score library project. `https://en.wikipedia.org/wiki/International_Music_Score_Library_Project`.

[10] MuseScore. Musescore. `https://musescore.org`, 2024. Version 4.3.0.

[11] Avid Technology. Sibelius. `https://www.avid.com/sibelius`, 2024. Version 2024.3.

[12] MakeMusic. Finale. `https://www.finalemusic.com/`, 2024.

[13] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, June 2022.

[14] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[16] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. Laion-5b: An open large-scale dataset for training next generation image-text models, 2022.

[17] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[18] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip.

[19] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2818–2829, 2023.

[20] Alec Radford, Jong Wook Kim, Chris Hallacy, A. Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[21] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.

[22] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[23] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium, 2018.

[24] Maximilian Seitzer. pytorch-fid: FID Score for PyTorch. https://github.com/mseitzer/pytorch-fid, August 2020. Version 0.3.0.

[25] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models, 2022.

[26] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[29] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[30] Soroush Mehri, Kundan Kumar, Ishaan Gulrajani, Rithesh Kumar, Shubham Jain, Jose Sotelo, Aaron Courville, and Yoshua Bengio. Samplernn: An unconditional end-to-end neural audio generation model. *arXiv preprint arXiv:1612.07837*, 2016.

[31] Andrea Agostinelli, Timo I Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, et al. Musiclm: Generating music from text. *arXiv preprint arXiv:2301.11325*, 2023.

[32] Ke Chen, Yusong Wu, Haohe Liu, Marianna Nezhurina, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. Musicldm: Enhancing novelty in text-to-music generation using beat-synchronous mixup strategies. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1206–1210. IEEE, 2024.

[33] Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark D Plumbley. Audioldm: Text-to-audio generation with latent diffusion models. *arXiv preprint arXiv:2301.12503*, 2023.

[34] Haohe Liu, Yi Yuan, Xubo Liu, Xinhao Mei, Qiuqiang Kong, Qiao Tian, Yuping Wang, Wenwu Wang, Yuxuan Wang, and Mark D Plumbley. Audioldm 2: Learning holistic audio generation with self-supervised pretraining. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024.

[35] Max WY Lam, Qiao Tian, Tang Li, Zongyu Yin, Siyuan Feng, Ming Tu, Yuliang Ji, Rui Xia, Mingbo Ma, Xuchen Song, et al. Efficient neural music generation. *Advances in Neural Information Processing Systems*, 36, 2024.

[36] Zhengcong Fei, Mingyuan Fan, and Junshi Huang. Music consistency models. *arXiv preprint arXiv:2404.13358*, 2024.

[37] Geoffrey Cideron, Sertan Girgin, Mauro Verzetti, Damien Vincent, Matej Kastelic, Zalán Borsos, Brian McWilliams, Victor Ungureanu, Olivier Bachem, Olivier Pietquin, et al. Musicrl: Aligning music generation to human preferences. *arXiv preprint arXiv:2402.04229*, 2024.

[38] 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, 2018.

[39] Aashiq Muhamed, Liang Li, Xingjian Shi, Suri Yaddanapudi, Wayne Chi, Dylan Jackson, Rahul Suresh, Zachary C Lipton, and Alex J Smola. Symbolic music generation with transformer-gans. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 408–417, 2021.

[40] Yi-Jen Shih, Shih-Lun Wu, Frank Zalkow, Meinard Müller, and Yi-Hsuan Yang. Theme transformer: Symbolic music generation with theme-conditioned transformer. *IEEE Transactions on Multimedia*, 25:3495–3508, 2022.

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

[42] 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.

A   MusicScore dataset

A.1   Dataset documentation

MusicScore dataset is an image-text dataset. For the image and text of every single sample, the image is the music score page of classical music, the text is the metadata of the corresponding composition which this page belongs to. MusicScore dataset varies in three different scales, 403, 14656, and 204800 pairs of image and text data, namely, MusicScore-400, MusicScore-14k and MusicScore-200k respectively.

As MusicScore is an image-text pair dataset, the motivation of crafting MusicScore dataset is to encourage the connection between visual musical semantics and text description, which currently is still an unexplored territory. By exploring the potentials on the visual musical semantic information, equipping with appropriate usage of it may enhance music generation, even music understanding field in some degree.

Among the smallest subset MusicScore-400, there are 3 complete piano pieces and 16 violin compositions. After partitioning the scores into separate images, the piano pieces consist of 215 pages, while the violin compositions consist of 188 pages. The metadata format of the MusicScore-400 dataset is consistent with MusicScore-14k and -200k.

MusicScore has been uploaded to Hugging Face: `https://huggingface.co/datasets/ZheqiDAI/MusicScore`. Three subsets in different scales are maintained, along with the Croissant metadata. For the scripts of color depth filtering and the PyTorch model code of score classification has also been uploaded to GitHub: `https://github.com/dzq84/MusicScore-script`. Additionally, we also open source the inference code of music score generation experiment: `https://github.com/Rozenthegoat/MusicScore-Inference/`.

A.2   Dataset processing

Among those 148,257 PDF music scores, not all of them are of high-quality for music score generation. The music scores in IMSLP vary in quality. First, many handwritten scores are manuscripts written by composers who lived centuries ago. Second, many PDF files contain catalogs, textual descriptions, and other front pages that are not relevant to music scores. Third, there are also a number of scores that have been manually scanned and uploaded, containing page misalignment errors or unclear symbols due to low scanner resolution. Figure 4 shows examples of two low-quality and two high-quality scores, respectively. We designed the following criteria to clean the music scores.

- **High Quality** High-quality images are usually created by publishers using professional tools or software such as MuseScore [10], Sibelius [11], and Finale [12]. The color of high-quality scores should be in grayscale with three RGB channels.

- **Full Page Score** In each page of a music score PDF, the proportion of musical content should occupy more than two-thirds of the page. A page should not contain too many texts irrelevant to the music content, such as those on the front cover page.

Non-score classifying model   The classifier is a ResNet-based model with a binary softmax as the last layer. To train the classifier, we manually labelled 450 images, including 270 musical score images and 180 non-musical score images. Additionally, we manually label 50 images for testing purposes, including 30 musical score images and 20 non-musical score images. We resize the music scores to the resolution of 2048×2048, feeding them into a ResNet18 [27] with a cross-entropy loss to distinguish the score image from non-score images.

A.2.1   Manually labelled subset

Training with large-scale datasets such as MusicScore-14k and -200k can be computationally intensive, not all users have access to sufficient computational resources. In order to lower the barrier for using MusicScore, we manually curated a small-scale dataset called MusicScore-400, consisting of 403 image-text pairs. The MusicScore-400 subset is not part of the MusicScore-200k. Instead, MusicScore-400 contains music scores partitioned from 19 compositions in history, composed by renowned musicians such as Bach, Beethoven, and Chopin. Among them, there are 3 complete piano

Figure 4: The first column: low-quality score with yellow pages. The second column: low-quality score with too much white spaces and noisy point. The third and fourth columns: high-quality scores.

pieces and 16 violin compositions. After partitioning the scores into separate images, the piano pieces consist of 215 pages, while the violin compositions consist of 188 pages. The metadata format of the MusicScore-400 dataset is consistent with MusicScore-14k and -200k. By providing this smaller dataset, we aim to make it more accessible for researchers with limited resources to explore and experiment with the music score image data, encouraging a wider range of engagement in tasks related to music score images and music generation.

A.3 Data statistics

We conducted a statistical analysis of MusicScore based on the metadata of music pieces. All statistics are calculated from the 32,307 PDF files. Figure 5 shows the statistics of music styles, instruments, keys, and composer time periods of the music scores. The top-left figure shows that the majority of piece styles are Modern, including 14,262 PDF files. The top-right figure shows that MusicScore contains mostly piano scores with 8,095 PDF files. Since there can be multiple instruments in one music score, the number of other instruments is 40,435, larger than the total number of 32,307 PDF files. The bottom-left figure shows the top seven keys of the music scores. The top three keys are C major, G major, and F major. The bottom-right figure shows the composer period. There are 17,414 Modern composers in the dataset, followed by 3,737 Renaissance composers and 3,395 Romantic composers. The MusicScore dataset does not have negative impacts on society and can be freely used by the public under CC BY 4.0 standard.

A.4 Intended use

An example sample refers to Figure 2. The image is a single page of classical music score. The text can be organised by different attributes of metadata which recorded as a JSON file in terms of different scenarios and usages. Instead of providing processed sentence or paragraph, the reason of providing raw metadata as the text part is to maximize the ways of usage. The intended tasks where MusicScore dataset can be applied are text-to-image score generation, contrastive learning for music score images and text description. For music understanding, MusicScore can be used in a classification task, contrastive learning task between score image and text description.

B Experimental

B.1 Latent Diffusion Model

VAE We apply variational autoencoder (VAE) [14] to encode the training images. The encoder E encodes $x \in \mathbb{R}^{H \times W \times 3}$ in RGB space, into a latent representation $z = E(x)$. The latent representation z has a shape of $[C, \frac{H}{f}, \frac{W}{f}]$, where f represents the downsampling factor. In [13], the suggested $f \in [4, 16]$ yields the best results. Therefore, we chose the same value of $f = 8$ as the Stable Diffusion [13] official implementation.
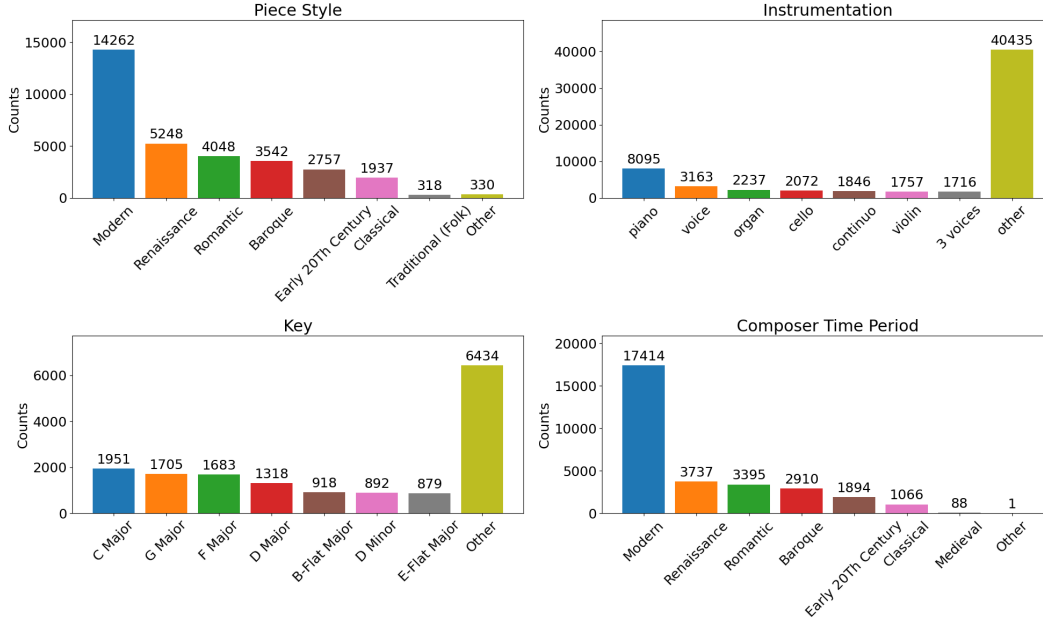
Figure 5: The data statistics of music scores in the MusicScore-200k dataset.

Text encoder In order to achieve our objective of guided generation using text, we utilize a text encoder to encode the text into embeddings that can serve as input to the model. The text encoder we use is OpenCLIP [18] in Stable Diffusion 2.0 [13] which an open source implementation of CLIP [20] by OpenAI. CLIP learns a multi-modal embedding space by jointly training an image encoder and text encoder to maximize the cosine similarity of the image and text embeddings [20]. These embeddings serve as conditions to drive the UNet generation backbone.

Text-driven latent diffusion model After calculating the text and image embeddings, UNet can be trained [15, 13] as the generation engine in this system. We employ OpenCLIP [18, 20] to transform text into embeddings. The text-derived embeddings are fed into the UNet via spatial cross-attention mechanism [13, 28] for denoising generation process, yielding the latent representation of the corresponding music score images. This latent representation is then input into VAE decoder to obtain the final score image. This constitutes the complete text-to-score image process, playable score images can be generated through this process.

B.2 Training setup

Training data There are two phases of fine-tuning in the experiment, fine-tuning VAE and UNet respectively. For fine-tuning VAE, the crafted dataset contains 240 images consist of 120 images of *Preludes and Fugues through all tones and semitones* by Johann Sebastian Bach which also occurs in MusicScore-400 subset that used in fine-tuning UNet, and another 120 violin score images which are randomly selected from MusicScore-14k subset. For fine-tuning UNet, the dataset is MusicScore-400 subset that contains 403 images.

Hyperparameters The hyperparameters used in the experiment is described in Table 2

Distributed training We use Distributed Data Parallel training throughout the whole experiment. The distribution is conducted by using Hugging Face's Accelerate, a library that simplifies distributed training of deep learning models, providing easy-to-use abstractions and utilities for efficient utilization of hardware resources. We use default Accelerate configuration which actually is a Data Distributed Parallel method. The training is conducted in FP32, without mixed precision training.

We use single machine with 8 RTX 4090 GPUs both VAE and UNet fine-tuning. The loss curve for fine-tuning UNet refer to Figure 6.

11

UNet Configuration	
Latent Dimension C_z	4
Downsample Factor f	8
Latent Scaling Factor	0.18215
Hyperparameters	
Learning Rate	0.00001
Batch Size	8
Adam ϵ	1e-8
Adam(β_1, β_2)	(0.9, 0.999)
Adam λ	0.01

Table 2: Detailed experimental settings.

For diffusion, three prediction modes are optional for Stable Diffusion 2.0, which are x_0-prediction, ϵ-prediction, and v-prediction. The v-prediction mode fuses prediction of x_0 and ϵ which considered as the state-of-the-art prediction method in many applications. In our experiment, we use ϵ-prediction mode, i.e., to predict noise.

B.3 Experiment result

Figure 7 illustrates the denoising process of text-generated score image. We use DDIM sampler with a DDIM denoising steps of 250 to perform denoising process.

Figure 8 illustrates the text-generated score images at 512×512 resolution.
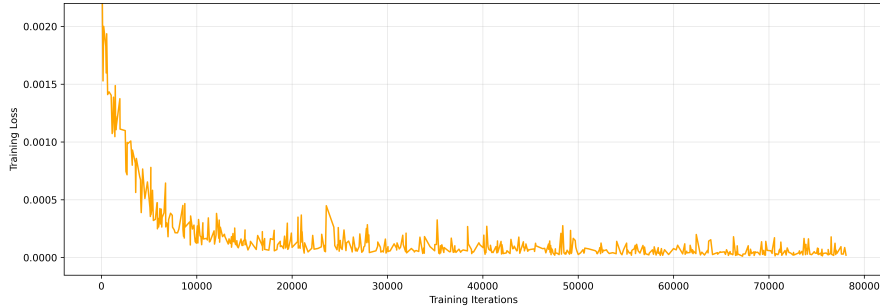

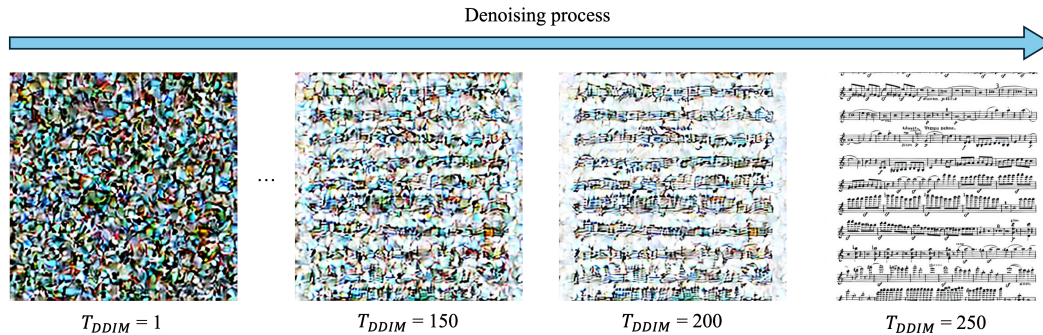Figure 6: Training loss curve for fine-tuning UNet.

Figure 7: **Example of denoised intermediate images in denoising process.** The denoising sampler is DDIM Sampler with 250 DDIM sampling steps. The intermediates are saved every 50 steps. All results are guided using *classifier-free guidance* of 4.0 *cfg* scale.

12

(a) a music score, instrumentation is violin, key is A major

(b) a music score, instrumentation is violin

(c) a music score, instrumentation is piano, key is A major

(d) a music score, instrumentation is piano

Figure 8: Generated music scores. The selected samples are generated by the text-driven Stable Diffusion model fine-tuned on MusicScore-400 at 512×512. For all results, the *classifier-free guidance* scale remains 4.0 and 250 DDIM sampling steps. The input text prompt follows the pattern of: *a music score, instrumentation is [instrumentation], key is [key]*.

C   Related works of music generation

Previously, there have been two major research directions to address the music generation problem, including audio music generation and symbolic music generation.

Audio music generation   Audio music generation systems generate music waveforms and audio representations. These systems can be trained end-to-end to produce audio signals without the need for intermediate symbolic music representations. WaveNet [29] is an representative music generation system modeled by dilated convolutional neural networks (CNNs). SampleRNN [30] applies an improved hierarchical recurrent neural network (RNN) to model waveforms. Recently, large language models have been applied to music generation, such as MusicLM [31]. Diffusion models are types

of probabilistic generative model that incrementally transforms noise into music through a series of learned denoising steps. Diffusion model-based music generation systems include MusicLDM [32], AudioLDM [33, 34], MeLoDy [35], and music consistency models [36]. MusicRL [37] aligns music generation with human preferences. The advantage of audio music generation systems lies in their universality. However, using audio music generation systems to generate long music waveforms may result in missing semantic information and is not easy to edit and manipulate.

Symbolic music generation Symbolic music generation [6] generate music in symbolic formats, such as MIDI and ABC notations. The music symbols include the instrument pitch, onset, velocity, and duration information of music. In symbolic music generation, the music symbols are transformed into a series of tokens and input to a language model. Music Transformer [38] is the first work to apply Transformer [28] for symbolic piano music generation. Transformers-based methods have been widely used for symbolic music generation in [39, 40]. Diffusion models are applied for symbolic music generation in [41]. MuseGAN [42] is a multiple-track music generation system that can generate four bars of music. The generated symbolic music has the advantage of being easy to edit, transpose, and manipulate. However, symbolic music data generation systems lack expressive information of music and require high-quality symbolic music data for training. The generated music symbols also lack performance articulations, which are important for musicians to play.

D   Limitations

The MusicScore dataset consists of 32,307 PDF files, which were selected from a pool of 148,257 PDF files to ensure quality, resulting in a selection rate of 21.8%. Consequently, some music scores from the original IMSLP pages may not be included in MusicScore. Second, MusicScore divides PDF files into individual JPG pages. The metadata describes the entire PDF file rather than individual JPG pages, potentially leading to mismatches between text and images. Third, the MusicScore dataset predominantly includes works by European composers, potentially creating an imbalance in the distribution of music works within the dataset.