Appendix for: "Multi-Aspect Conditioning for Diffusion-Based Music Synthesis: Enhancing Realism and Acoustic Control"

Ben Maman[⋆] Johannes Zeitler[⋆] Meinard Müller[⋆] Amit H. Bermano[†]

[⋆]International Audio Laboratories Erlangen, Germany
[†] Tel Aviv University

I. Dataset Information

We provide here the distribution of ensembles (instrumentation) and versions in the train set, the distribution of ensembles in the evaluation set, and the list of pieces used for the listening tests.

Figure 8 shows the distribution of ensembles in the train set $\mathcal{D}_{\mathrm{train}}$, which corresponds to the information in Table II in the main paper. Figure 9 shows the ranges of version lengths for the different ensembles in Figure 8 containing more than a single version. It can be seen that version lengths vary between a few minutes (e.g. guitar or organ in Figure 9), and a few hours (e.g. violin and piano duo, or violin, cello and piano trio in Table II in the main paper). Note that the version lengths appearing in Figure 9 are not the same as the total ensemble durations appearing in Figure 8. For example, individual guitar versions include no more than a few minutes (Figure 9), but the total amount of guitar recordings is almost five hours (Figure 8).

We note that acoustic control can be obtained in an easy and straightforward manner through conditioning on the version and the instrumentation, despite imbalance in ensembles and version lengths in the train set. Samples on the project page demonstrate the ability to successfully condition the model on versions with recordings of a few minutes—the guitar and organ versions, as well as the orchestral version "Czech Symphony Orchestra playing Beethoven's Coriolan Overture" each contain less than ten minutes.

The training data $\mathcal{D}_{\mathrm{train}}$ includes both high-quality audio of relatively new recordings acquired from Youtube,[1] and vintage recordings mainly from Museopen.[2] This was done in order to enable reproducing both unique vintage sound, and more modern high-quality sound. Although vintage recordings may contain artifacts due to old recording equipment, audio copying, editing and so on, we regard this as part of the version condition, which implicitly encompasses the recording quality.

The recordings we picked were of pieces for which corresponding MIDIs were available on KunstDerFuge,[3] in order to be able to train the transcriber used for conditioning the synthesizer. The MIDIs were verified to be on the same musical scale as the audio. In rare cases when they were not on the same musical scale, the MIDIs were pitch-shifted to match the audio's musical scale. This occurred mainly in Baroque music, where different tuning systems can deviate by a semitone or tone from the standard $440\mathrm{Hz}$ tuning.

TABLE IX
Composer distribution of the train set $\mathcal{D}_{\mathrm{train}}$.

Ensemble	Length	#Versions
Albeniz	1:54:57	25
Bach	21:52:41	54
Beethoven	14:53:21	12
Brahms	2:13:48	6
Cambini	0:33:10	1
Chopin	1:09:31	2
Mendelssohn	0:39:23	1
Mozart	4:56:50	19
Saint Saëns	0:20:11	1
Schubert	3:06:31	6
Sibelius	0:51:58	5
Sor	1:17:06	27
Tarrega	1:28:10	33
Tchaikovsky	3:08:10	5
All	58:25:47	197

Table IX shows the distribution of composers in $\mathcal{D}_{\mathrm{train}}$.

Table X shows the pieces appearing in the evaluation set used for the listening tests, which we refer to as $\mathcal{D}_{\mathrm{listen}}$, together with their instrumentation. Table XI shows the ensemble distribution of the large evaluation set used for quantitative evaluation, which we refer to as $\mathcal{D}_{\mathrm{quant}}$.

II. Quantitative Results over Large Scale Evaluation Set

In this section, we compare different models trained on the $\mathcal{D}_{\mathrm{train}}$ dataset. We perform the evaluation on the large evaluation set $\mathcal{D}_{\mathrm{quant}}$ appearing in Table XI.

We compare the following models: U-Net and T5 models trained with alignments as score conditions (these two models appear in our previous work [1]), and a T5 model trained with automatic transcriptions as score conditions (evaluation in this paper was done using this model). For the latter model we provided two forms of score conditions: Pitch with instrument, and pitch-only (see project page for samples generated with pitch-only input). The transcriptions were obtained by training a transcriber [2, 3] on the pairs of audio and alignments, and

[1] https://www.youtube.com
[2] https://musopen.org
[3] https://www.kunstderfuge.com

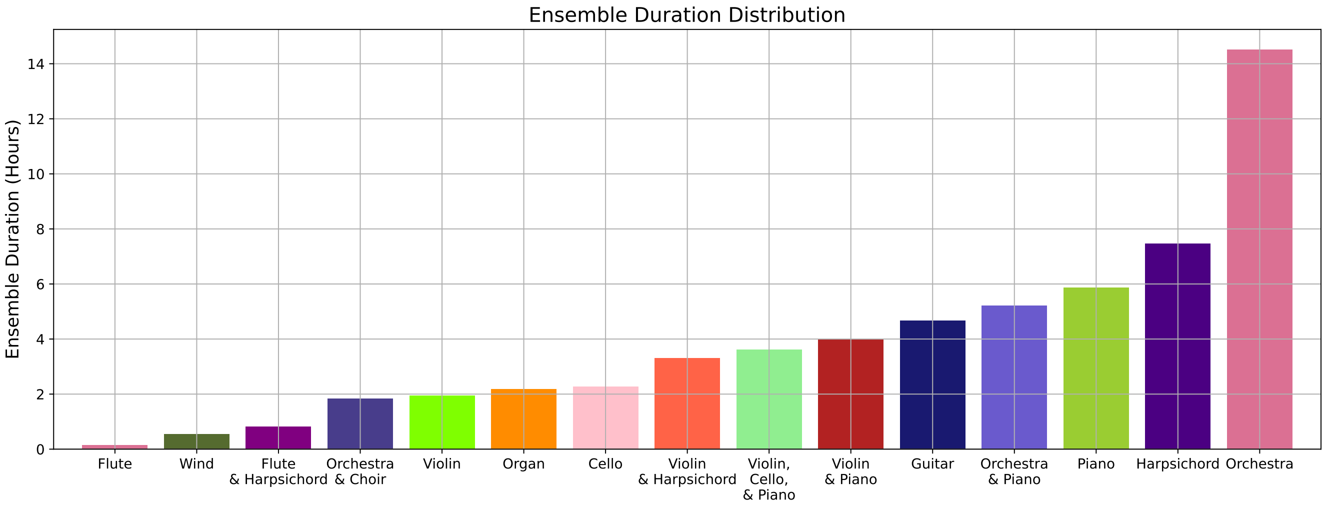

Fig. 8. Duration of ensembles in the train set by ensemble, in hours. This plot corresponds to Table II in the main paper.

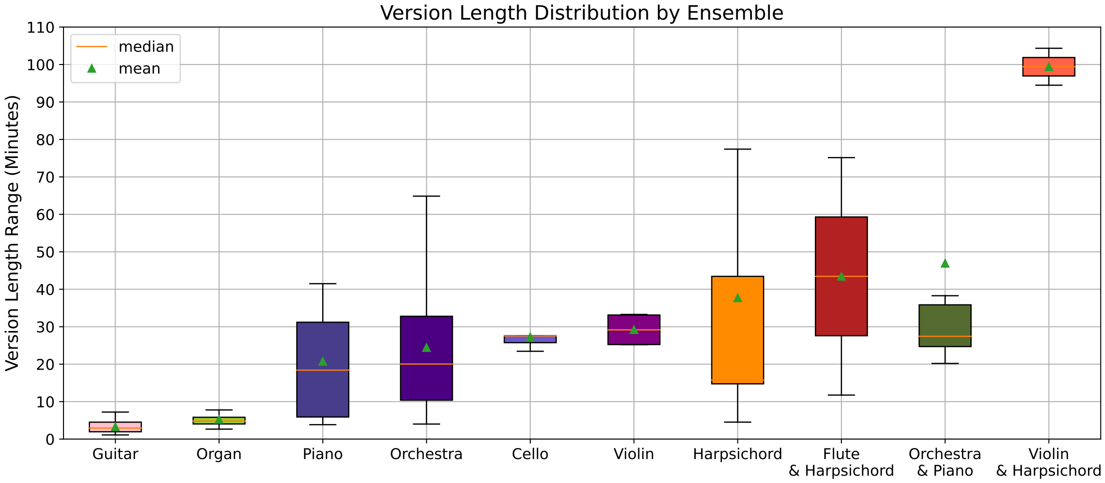

Fig. 9. Distribution of version lengths in the train set by ensemble, in box-plot form. Each box corresponds to an ensemble (i.e. instrumentation), and depicts the range of lengths of different versions in the train set for that ensemble. Ensembles from Table II in the main paper and Figure 8 that contain a single version do not appear in this figure since their distribution is trivial.

TABLE X

MIDI PERFORMANCES USED FOR THE LISTENING TEST WITH THEIR CORRESPONDING ENSEMBLES. THE FIRST THREE MINUTES OF EACH PERFORMANCE WERE USED. THIS DATASET IS REFERRED TO IN THE PAPER AS $\mathcal{D}_{\text{listen}}$. IT CONTAINS BOTH MIDI AND CORRESPONDING AUDIO RECORDINGS OF REAL MUSICAL PERFORMANCES, OF THE EXACT SAME ENSEMBLES, OF PIECES AND VERSIONS THAT DO NOT APPEAR IN THE TRAIN SET.

MIDI	**Ensemble**
Bach Double Concerto in C Minor BWV 1060 (movement 1)	violin, cello, & harpsichord
Bach Great Fugue in G Minor BWV 542	harpsichord
Bach Italian Concerto in F Major BWV 971 (movement 1)	harpsichord
Bach Mass in B Minor BWV 232, Gloria - Cum Sancto Spiritu (movement 11)	choir & orchestra
Bach Orchestral Suite 1 in C Major BWV 1066, Overture	orchestra
Bach Orchestral Suite 2 in B Minor BWV 1067, Badinerie	wind quintet
Bach Toccata and Fugue in D Minor BWV 565	church organ
Beethoven Symphony 5 in C Minor Op. 67 (movement 1)	orchestra
Beethoven Symphony 6 in F Major Op. 68 (movement 1)	orchestra
Beethoven Symphony 6 in F Major Op. 68 (movement 3)	orchestra
Mozart Piano Concerto 20 in D Minor K. 466 (movement 1)	piano & orchestra
Mozart Symphony 40 in G Minor K. 550 (movement 1)	orchestra

providing the thresholded predictions as note conditions to the synthesizer, rather than the alignments.

Together, 4 different configurations were evaluated. We train all models both with, and without version conditioning – together 8 configurations.

Results are reported in Tables XII (All-FAD), XIII (Group-FAD), XIV (version classification accuracy), and XV (transcription accuracy). The general trend we clearly and con-

TABLE XI
Instrument distribution of the large evaluation set $\mathcal{D}_{\text{quant}}$, used for quantitative evaluation. This dataset contains MIDI only, without corresponding audio. We show the total length for each ensemble (Length), and how many MIDI performances are played by each ensemble (#Performances).

Ensemble	**Length**	**#Performances**
Flute & Harpsichord	0:04:11	1
Orchestra	2:16:02	19
Orchestra & Choir	0:11:49	4
Orchestra & Piano	0:52:27	3
Solo Cello	0:05:53	3
Solo Guitar	0:05:07	3
Solo Harpsichord	0:09:17	4
Solo Organ	0:07:33	1
Solo Piano	0:18:53	2
Solo Violin	0:13:14	5
Violin & Harpsichord	0:05:44	2
Violin, Cello, & Piano	0:34:29	9
Wind Quintet	0:04:51	2
All	5:09:30	58

TABLE XII
All-FAD results on large evaluation set $\mathcal{D}_{\text{quant}}$.

	All-FAD↓			
	VGGish		TRILL	
Version Cond.	w/o	w/	w/o	w/
U-Net Aligned	3.37	3.94	0.12	0.11
T5 Aligned	3.98	3.53	0.12	0.09
T5 Transcribed	3.05	3.58	0.12	0.11
T5 Transcribed Pitch	3.97	3.78	0.18	0.12

TABLE XIII
Group-FAD results on large evaluation set $\mathcal{D}_{\text{quant}}$.

	Group-FAD↓			
	VGGish		TRILL	
Version Cond.	w/o	w/	w/o	w/
U-Net Aligned	7.06	5.21	0.5	0.33
T5 Aligned	6.95	5.46	0.51	0.35
T5 Transcribed	7.46	5.68	0.55	0.36
T5 Transcribed Pitch	12.81	6.03	0.7	0.38

sistently observe is that Group-FAD and version classification accuracy dramatically improve as a result of version conditioning, while All-FAD and transcription metrics remain comparable (possibly with a slight increase or decrease). That is, by incorporating version conditioning, we can generate performances with the same notes, but such performances that more resemble the desired target versions, and do so while maintaining quality.

We also notice that training with transcription-based score conditions yields comparable results to alignment-based score conditions, with transcription-based conditions producing slightly better transcription scores, and alignment-based conditions producing slightly better FAD scores.

We describe the results in detail:

a) All-FAD: One can see in Table XII that version conditioning does not significantly impact the All-FAD – the VGGish All-FAD slightly increases while the TRILL All-FAD slightly decreases. Increase in All-FAD as a result of version conditioning is not necessarily surprising—it can be interpreted as follows: Version conditioning causes the generated performances to deviate from the general distribution of the the train set $\mathcal{D}_{\text{train}}$, towards the distribution of a subset $\mathcal{D}_v$ corresponding to a specific version v.

The different compared models produce comparable results, except for the pitch-only model. It can be seen that conditioning on instrumentation improves the All-FAD score.

b) Group-FAD, Version Classification: The dramatic effect of version conditioning can be seen in Tables XIII and XIV. The Group-FAD metric dramatically improves as a result of version conditioning, for all models, without exception. This means the distribution of the generated performances becomes perceptually more similar to the conditioning versions.

c) Transcription: We observe the transcription metrics (Table XV), measuring if the synthesized performances actually realize the notes specified by the MIDI note condition. There is no entirely objective or absolute way to measure this, however, we can still gain insights by using an automatic transcriber. We therefore use a transcriber trained on precisely the same data as the synthesizer. Note that such metrics are influenced not only by the quality of the synthesizer, but also from the quality of the transcriber.

In Table XV we can observe that most models yield similar transcription metrics, whether using version conditioning or not, reaching accuracy of up to 67% (note-level), which is of reasonable magnitude when considering the complexity of highly polyphonic orchestral music. In addition, we can observe that using transcriptions as conditions rather than alignments provides better overall transcription metrics.

It can also be seen that unsurprisingly, the note-with-instrument metric is significantly lower when using pitch-only input. Note however that this is significantly mitigated when using version conditioning (15% improvement). This indicates that version conditioning helps achieving the target instrumentation, even when using pitch-only note conditions, as the model learns the correlations between versions and their instrumentation. This is further supported by the qualitative samples on the project page generated from pitch-only input with different version conditions.

A. Difference in All-FAD vs. Group-FAD Value Range

When comparing the actual FAD values in Tables XII, XIII, one can see the All-FAD values are lower than the Group-FAD. We attribute this to the fact that the mean vectors and covariance matrices for All-FAD are computed over significantly larger evaluation and reference datasets than for Group-FAD and therefore yield less statistical fluctuations, resulting in an overall lower FAD score.

III. Classifier-Free Guidance with Multiple Conditions

Classifier-Free Guidance (CFG) [4] is a technique for controlling the condition strength in conditional diffusion models. The model is trained both conditionally and unconditionally simultaneously, by applying condition dropout (zeroing out the condition), typically with probability 0.1. During sampling,

TABLE XIV
VERSION CLASSIFICATION RESULTS ON LARGE EVALUATION SET $\mathcal{D}_{\text{quant}}$.

	Classification% Top-1/3/5↑	
Version Cond.	w/o	w/
U-Net Aligned	11.6/24.5/36.1	52.9/73.5/84.5
T5 Aligned	35.5/60.0/69.7	67.7/89.7/91.0
T5 Transcribed	16.8/31.6/41.9	66.5/83.9/88.4
T5 Transcribed Pitch	4.5/18.1/27.1	62.6/79.4/85.8

TABLE XV
TRANSCRIPTION RESULTS ON LARGE EVALUATION SET $\mathcal{D}_{\text{quant}}$.

	Transcription F1% ↑					
	Note		Note & Inst.		Frame	
Version Cond.	w/o	w/	w/o	w/	w/o	w/
U-Net Aligned	63.1	61.8	46.6	46.2	65.4	64.1
T5 Aligned	62.0	63.2	38.4	47.1	60.7	62.0
T5 Transcribed	66.9	64.7	50.7	46.2	64.8	63.9
T5 Transcribed Pitch	67.2	64.3	25.0	40.3	64.7	63.2

noise is predicted both with and without the condition, and the enhanced conditioning is obtained though extrapolation in the condition's "direction":

$$\begin{aligned} \epsilon_{\text{cond}} &= \epsilon_\theta(x_t, t, c) \\ \epsilon_{\text{uncond}} &= \epsilon_\theta(x_t, t, 0) \\ \epsilon &= \epsilon_{\text{cond}} + (w-1)(\epsilon_{\text{cond}} - \epsilon_{\text{uncond}}) \end{aligned} \tag{1}$$

where $w > 1$ is an extrapolation weight controlling the desired conditioning strength. In our case of multi-aspect-conditioned music synthesis, we apply two simultaneous conditions: A score condition defining the notes to be played, and a version condition defining the acoustic- and performance-related properties such as timbre, recording environment, style, etc. It is possible to apply CFG using multiple conditions in a straightforward manner: Denote by c_1, c_2 the two types of conditions. During training, we dropout each of the conditions c_1, c_2 with probability 0.1, independently. Then, during sampling, we define:

$$\begin{aligned} \epsilon_{\text{cond1}} &= \epsilon_\theta(x_t, t, c1, 0) \\ \epsilon_{\text{cond2}} &= \epsilon_\theta(x_t, t, 0, c2) \\ \epsilon_{\text{cond1,2}} &= \epsilon_\theta(x_t, t, c1, c2) \\ \epsilon &= \epsilon_{\text{cond12}} + (w_1-1)(\epsilon_{\text{cond1,2}} - \epsilon_{\text{cond2}}) \\ &\quad + (w_2-1)(\epsilon_{\text{cond1,2}} - \epsilon_{\text{cond1}}) \end{aligned} \tag{2}$$

where $w_1, w_2 > 1$ are extrapolation weights representing the strengths of the two conditions, respectively. Intuitively, the term $(w_1 - 1)(\epsilon_{cond1,2} - \epsilon_{cond2})$ enhances the condition c_1 while remaining faithful to the condition c_2, and the term $(w_2 - 1)(\epsilon_{cond1,2} - \epsilon_{cond1})$ enhances the condition c_2 while remaining faithful to the condition c_1. We used extrapolation weights of $w_1 = w_2 = 1.25$ for score and version after a grid search with values $1.0, 1.25, 1.5, 2.0$ for w_1 and w_2. We note that other ways to perform the extrapolation are possible, such as using $\epsilon_\theta(x_t, t, 0, 0)$ (zeroing out both conditions), but the approach described in Equation 2 gave best results in practice.

IV. U-NET TRAINING

Information on the T5 Transformer training appears in the main paper. For the U-Net, we use a similar architecture to that of Ho et al. [5]. We adapt it to spectrogram diffusion by using 1D operations (convolution, attention and group normalization). We use 304 feature dimensions in the first layer, which increase by a factor of 2 in each block. Similar training to that of the T5 took approximately 100 hours on two Nvidia A100 GPUs.

REFERENCES

[1] B. Maman, J. Zeitler, M. Müller, and A. H. Bermano, "Performance conditioning for diffusion-based multi-instrument music synthesis," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Seoul, South Korea, 2024, pp. 5045–5049.

[2] 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," in *Proceedings of the International Conference on Learning Representations (ICLR)*, New Orleans, Louisiana, USA, 2019. [Online]. Available: https://openreview.net/forum?id=r1lYRjC9F7

[3] B. Maman and A. H. Bermano, "Unaligned supervision for automatic music transcription in the wild," in *Proceedings of the International Conference on Machine Learning (ICML)*, Baltimore, Maryland, USA, 2022, pp. 14 918–14 934.

[4] J. Ho and T. Salimans, "Classifier-free diffusion guidance," *arXiv*, vol. abs/2207.12598, 2022. [Online]. Available: https://doi.org/10.48550/arXiv.2207.12598

[5] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems*, 2020. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html