An Independence-promoting Loss for Music Generation with Language Models

Jean-Marie Lemercier [1 † *] **Simon Rouard** [2 3 *] **Jade Copet** [3] **Yossi Adi** [3 4] **Alexandre Defossez** [5 †]

Abstract

Music generation schemes using language modeling rely on a vocabulary of audio tokens, generally provided as codes in a discrete latent space learnt by an auto-encoder. Multi-stage quantizers are often employed to produce these tokens, therefore the decoding strategy used for token prediction must be adapted to account for multiple codebooks: either it should model the joint distribution over all codebooks, or fit the product of the codebook marginal distributions. Modelling the joint distribution requires a costly increase in the number of auto-regressive steps, while fitting the product of the marginals yields an inexact model unless the codebooks are mutually independent. In this work, we introduce an independence-promoting loss to regularize the auto-encoder used as the tokenizer in language models for music generation. The proposed loss is a proxy for mutual information based on the maximum mean discrepancy principle, applied in reproducible kernel Hilbert spaces. Our criterion is simple to implement and train, and it is generalizable to other multi-stream codecs. We show that it reduces the statistical dependence between codebooks during auto-encoding. This leads to an increase in the generated music quality when modelling the product of the marginal distributions, while generating audio much faster than the joint distribution model.

1. Introduction

Generative models are being increasingly used to produce multimedia content such as e.g. image (Rombach et al., 2022), text (Brown et al., 2020), speech (van den Oord et al., 2016; Kong et al., 2020; 2021) or audio (Borsos et al., 2023; Agostinelli et al., 2023; Yang et al., 2023b; Kreuk et al., 2023). These models rely on artificial neural networks parameterizing approaches such as generative adversarial networks (Goodfellow et al., 2014), diffusion models (Ho et al., 2020; Song & Ermon, 2019) or transformer-based language models (Radford et al., 2019; Vaswani et al., 2017). We focus here on the task of generating music based on a text prompt. Music signals occupy the full frequency spectrum (unlike speech) and can be very long sequences (unlike most images), making the generation task arduous. Text-to-music language models (Agostinelli et al., 2023; Kreuk et al., 2023; Copet et al., 2023; Borsos et al., 2023) try to model the distribution of a vocabulary of discrete units i.e. tokens. The audio tokens are often generated by a multi-stage quantizer operating in the latent space learnt by a neural compression model (Défossez et al., 2023; Zeghidour et al., 2021). As the quantizer uses a distinct codebook for each stage, the language model decoding strategy must be adapted to model either the joint distribution over all codebooks, or the factorization of codebook marginal distributions. On the one hand, modelling the joint distribution requires either using an impractically large vocabulary size, or multiplying the number of auto-regressive timesteps by the number of codebooks. On the other hand, modelling the factorized distribution significantly facilitates the training of the language model and speeds inference up, but only provides an approximation of the true model. Several strategies for modelling the factorized distribution have been proposed (Wang et al., 2023; Kharitonov et al., 2022; Kreuk et al., 2023; Copet et al., 2023) yielding satisfying results. However, we argue that these solutions do not directly address the issue, which is that the factorized distribution is equivalent to the full joint distribution *only if the codebooks are mutually independent*.

In this work, we propose to introduce an independence constraint between codebooks, in the form of an auxiliary objective for training the auto-encoder used as the tokenizer for the language model. Instead of leveraging adversarial training as in (Belghazi et al., 2018; Brakel & Bengio, 2017), we propose to use a proxy for mutual information based on the maximum mean discrepancy (Gretton et al., 2012),

[*]Equal contribution [1]Universität Hamburg [2]IRCAM [3]Meta AI [4]The Hebrew University of Jerusalem [5]Kyutai [†]Work done while at Meta AI. Correspondence to: Jean-Marie Lemercier <jeanmarie.lemercier@uni-hamburg.de>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

which solves a dual formulation of earth mover optimization in Gaussian reproducible kernel Hilbert spaces. We conduct experiments on music generation, and run ablations with respect to our independence-promoting loss configurations.

We make the following contributions:

- We show that the maximum mean discrepancy in reproducible kernel Hilbert spaces is a reasonable proxy for independence, since optimizing our criterion leads to a reduction of mutual information between codebooks during auto-encoding.
- We propose a modified version of our loss that matches the decoding strategy used for token prediction. When applied to the "delay" strategy proposed in (Kharitonov et al., 2022), we obtain the best performance across all our models.
- We show that objective and subjective music generation quality scores favour the language model whose tokenizer was trained with the proposed independence loss in comparison to other baselines. Our resulting model has the same amount of parameters and generation speed as the baseline not using our proposed criterion. Our approach enables to generate audio at the same frame rate as the auto-encoder, which is much faster than the joint distribution model and has similar generation quality.

Please visit our companion website[1] for audio examples, support with code, etc.

2. Background

2.1. Quantization

Quantization is a discretization method aiming at reducing the bitrate used to encode information, which is a major challenge in low-resource communications. Quantization is also used in machine learning, typically to reduce the memory and computational footprints of deep neural networks (DNNs) on embedded devices. More recently, quantizers were used to produce a vocabulary of discrete units for language models learning the distribution of originally continuous signals such as e.g. images or audio. Quantization schemes can be categorized in two classes: scalar and vector quantization. Scalar quantization discretizes each dimension of the considered signal, rounding the current value to the closest bin on a quantization grid. Vector quantization (VQ) (Gray, 1984) encodes signals as entries (or *codes*) in a multi-dimensional codebook. Concretely, VQ learns a codebook $\mathcal{C}$ with M vectors of dimension N and at inference, it performs a nearest neighbour search in the codebook space to find the right code for the input signal.

Multi-stage vector quantizers (Juang & Gray, 1982; Vasuki & Vanathi, 2006) use multiple codebooks with reasonable size, which increases codebook utilization compared to having one large codebook. This is one of the keys to the success of these structured quantizers, which achieve a good trade-off between computational complexity and coding efficiency. Residual vector quantization (RVQ) (Zeghidour et al., 2021) is a multi-stage vector quantization scheme that introduces K codebooks. At each stage $k \in \{1, \dots, K\}$, the residual of the previous stage is quantized with the codebook $\mathcal{C}^{(k)}$ and the residual for the next stage is obtained by subtracting the resulting code from the previous residual. The codes exhibit a natural hierarchical, coarse-to-fine structure, as most of the information is contained in the first few codebooks.

2.2. Independence of Random Variables

Reliably measuring statistical dependence between random variables is a wide-spread topic in the machine learning literature (Higgins et al., 2017; Burgess et al., 2017; Brakel & Bengio, 2017; Hyvarinen et al., 2023; Belghazi et al., 2018). Let $\{Z_1, \dots, Z_K\}$ be a family of vector random variables in $\mathbb{R}^N$. It is an independent family if and only if the joint distribution, denoted as $\mathbb{P}_Z$, and the product of the marginal distributions denoted as $\mathbb{P}_{\bar{Z}}$ (or *factorized* distribution) coincide. This is equivalent to saying that the joint probability density function can be factorized into the product of the marginal probability density functions, i.e. $\forall J \leq K$, $\forall (k_1, \dots k_J) \in \{1, \dots, K\}^J$ with $i \neq j \Rightarrow k_i \neq k_j$ and $\forall (z_{k_1}, \dots z_{k_J}) \in \mathbb{R}^{N \times J}$:

$$p_{Z_{k_1}, \dots, Z_{k_J}}(z_{k_1}, \dots, z_{k_J}) = \Pi_{j=1}^{J} p_{Z_{k_j}}(z_{k_j}). \quad (1)$$

where p_X is the probability density function of the random variable X. Independence between variables can be exactly measured via the mutual information $\mathcal{I}(Z_1, \dots Z_K)$, which equals the Kullback-Leibler divergence between the joint distribution $\mathbb{P}_Z$ and the factorized distribution $\mathbb{P}_{\bar{Z}}$. This instance of mutual information is called *total correlation*, and can also be expressed in terms of entropies:

$$\mathcal{I}(Z_1, \dots Z_K) = \mathrm{D}_{\mathrm{KL}}\left(\mathbb{P}_Z || \mathbb{P}_{\bar{Z}}\right) \quad (2)$$

$$= \mathcal{H}(Z_1, \dots, Z_K) - \sum_{k=1}^{K} \mathcal{H}(Z_k), \quad (3)$$

where $\mathcal{H}(X)$ measures the entropy of the random variable X. While a closed-form computation of the total correlation is available through (3), this requires either exact knowledge of the distributions, or approximate knowledge through histogram estimation. We will eliminate the first option since we do not posit distributional assumptions as in e.g. the

[1] encodec-mmd.github.io

variational auto-encoder (VAE) case (Kingma & Welling, 2014; Higgins et al., 2017). Estimating the histogram of the marginal variables Z_i might be possible most of the time. However, estimating the histogram of the joint variable $(Z_1, \dots, Z_K)$ is a tedious operation as it requires an immense sample size. Another poor property of histograms is that their computation is not differentiable.

For the reasons listed above, we should resort to proxies to force the independence of random variables. Several independence proxies have already been proposed in the literature (Belghazi et al., 2018; Brakel & Bengio, 2017; Li et al., 2023). However, these often rely on adversarial training, which is known to significantly increase the training difficulty (Goodfellow et al., 2014). For instance (Belghazi et al., 2018) optimize a dual formulation of the Kullback-Leibler divergence through adversarial training of neural estimators. A similar paradigm was already explored for non-linear independence component analysis (ICA) (Hyvarinen et al., 2023), where a neural network was trained to discriminate between samples from the joint distribution and samples from the factorized distribution (Brakel & Bengio, 2017). A Jensen-Shannon divergence objective is then formulated and optimized using the estimated joint-to-factorized probability ratio (Huszar, 2016).

Aside the Kullback-Leibler and Jensen-Shannon divergences, another convenient distance between probability distributions is the earth mover distance, defined as:

$$W_2(\mathbb{P}_Z||\mathbb{P}_{\bar{Z}}) = \inf_{\pi \in \Pi(\mathbb{P}_Z, \mathbb{P}_{\bar{Z}})} \mathbb{E}_{(Z,\bar{Z})\sim\pi} \left\| Z - \bar{Z} \right\|_2, \quad (4)$$

where $\Pi(\mathbb{P}_Z, \mathbb{P}_{\bar{Z}})$ denotes the ensemble of all distributions whose marginals are $\mathbb{P}_Z$ and $\mathbb{P}_{\bar{Z}}$. Given the Kantorovic-Rubinstein duality (Villani, 2009), the earth mover distance coincides with the maximum mean discrepancy (MMD) (Gretton et al., 2012) defined as a simpler optimization problem over real-valued 1-Lipschitz functions:

$$\begin{aligned} \mathrm{MMD}(\mathbb{P}_Z||\mathbb{P}_{\bar{Z}}) &= W_2(\mathbb{P}_Z||\mathbb{P}_{\bar{Z}}) \\ &= \sup_{h, \|h\| \leq 1} \mathbb{E}_{Z\sim\mathbb{P}_Z}[h(Z))] - \mathbb{E}_{\bar{Z}\sim\mathbb{P}_{\bar{Z}}}[h(\bar{Z})]. \end{aligned} \quad (5)$$

Since MMD is equivalent to the earth mover distance, if $\mathrm{MMD}(\mathbb{P}_Z||\mathbb{P}_{\bar{Z}}) = 0$ then the joint distribution $\mathbb{P}_Z$ and the factorized distribution $\mathbb{P}_{\bar{Z}}$ are equal and therefore the family $\{Z_1, \dots, Z_K\}$ is independent.

One could use a neural network to parameterize the function h and train it with an adversarial loss, which would resemble the aforementioned works (Belghazi et al., 2018; Brakel & Bengio, 2017). This was applied in (Arjovsky et al., 2017), although for density estimation in generative adversarial networks (GANs) rather than independence optimization. However, (Gretton et al., 2012) highlight a remarkable property of the MMD by taking the set of functions h to be the unit ball in an reproducible kernel Hilbert space (RKHS) $\mathbb{H}$.

Let $X \in \mathbb{R}^{N\times J}$: an *evaluation operator* $\delta_X : \mathbb{H} \to \mathbb{R}$ associates $h \in \mathbb{R}$ to its evaluation $h(X) \in \mathbb{R}$. The Riesz representation theorem guarantees that for each continuous evaluation operator δ_X, there exists a *feature mapping* $\phi(X) \in \mathbb{H}$, such that $\forall h \in \mathbb{H}, \delta_X(h) := h(X) = \langle h, \phi(X)\rangle_{\mathbb{H}}$. A core property of RKHSs is that they are equipped with a kernel function $k : \mathbb{R}^{N\times J} \times \mathbb{R}^{N\times J} \to \mathbb{R}$, such that dot products between features can be conveniently computed as $\langle \phi(X), \phi(Y)\rangle_{\mathbb{H}} = k(X, Y)$. It can be then shown that a lower-bound of the MMD in (5) can be obtained as a combination of kernel computations:

$$\begin{aligned} \mathrm{MMD}_{\mathbb{H}}(\mathbb{P}_Z||\mathbb{P}_{\bar{Z}}) = \; & \mathbb{E}_{Z_1\sim\mathbb{P}_Z}\mathbb{E}_{Z_2\sim\mathbb{P}_Z} k(Z_1, Z_2) \\ + \; & \mathbb{E}_{\bar{Z}_1\sim\mathbb{P}_{\bar{Z}}}\mathbb{E}_{\bar{Z}_2\sim\mathbb{P}_{\bar{Z}}} k(\bar{Z}_1, \bar{Z}_2) \\ - \; & 2\mathbb{E}_{Z_1\sim\mathbb{P}_Z}\mathbb{E}_{\bar{Z}_2\sim\mathbb{P}_{\bar{Z}}} k(Z_1, \bar{Z}_2) \\ \leq \; & \mathrm{MMD}(\mathbb{P}_Z||\mathbb{P}_{\bar{Z}}). \end{aligned} \quad (6)$$

The proof is let to appendix A. An important property of $\mathrm{MMD}_{\mathbb{H}}$ is that if $\mathbb{H}$ is a *universal* RKHS, then $\mathrm{MMD}_{\mathbb{H}}(\mathbb{P}_Z||\mathbb{P}_{\bar{Z}}) = 0 \iff \mathbb{P}_Z = \mathbb{P}_{\bar{Z}}$ (Gretton et al., 2012). This shows that if we achieve optimality for our lower-bound $\mathrm{MMD}_{\mathbb{H}}$ using a universal RKHS, we actually obtain an independent representation. A RKHS $\mathbb{H}$ is said universal if it is dense in the space of functions $h : \mathbb{R}^{N\times J} \mapsto \mathbb{R}$. In particular, RKHSs with Gaussian kernels are universal.

Our proposed proxy can easily be computed with batch estimators and does not require adversarial training. Another kernel-based estimator was presented in (Li et al., 2023; Yu et al., 2021). However, it requires a singular-value decomposition of the kernel matrices $k(Z_1, Z_2)$ which is sensitive to numerical errors, produces gradients with high variance and is costly for high-dimensional data.

2.3. Audio Generation with Language Models

Language modelling using auto-regressive Transformer-style architectures (Vaswani et al., 2017) has been central in audio generation lately (Dhariwal et al., 2020; Borsos et al., 2023; Wang et al., 2023; Agostinelli et al., 2023; Kreuk et al., 2023; Copet et al., 2023). These approaches typically consist of two modules. The first is a neural audio compression model such as e.g. (Zeghidour et al., 2021; Défossez et al., 2023) that takes as input the raw audio $X \in \mathbb{R}^L$ with L the sequence length. The encoder part of this codec transforms X into a discrete token sequence with codebook indexes $Q \in \{1, \dots, M\}^{T\times K}$ and corresponding codes $Z \in \mathbb{R}^{T\times K\times N}$, where T is the reduced time length obtained via the encoder strides, K is the number of codebooks, M is the codebook size and N is the codebook dimension. The second module is an autoregressive

Transformer-decoder language model operating in the space of discrete audio tokens. Given a textual conditioning C provided by a pre-trained text encoder, the language model f_θ predicts the distribution of a sequence of tokens Z autoregressively as $f_\theta(Z^{(t)}|\, C, Z^{(1)}, \dots, Z^{(t-1)})$. Finally, the acoustic tokens generated by the language model are provided to the audio decoder to synthesize the final waveform.

Because VQ-based audio codecs typically use multiple codebooks for optimal compression, the usual single-stream decoding strategy of language models needs to be adapted. The token sequence can be for instance flattened, and the transformer then predicts the codebooks sequentially. Theoretically, this leads to modelling the joint distribution of codebooks $\mathbb{P}_Z$ (Copet et al., 2023). However, this approach yields high computational complexity as the frame rate is multiplied by the number of codebooks K compared to the auto-encoder.

Another solution is to decode the distributions of each codebook independently and thus modelling the factorized distribution $\mathbb{P}_{\bar{Z}}$ conditionally to the past tokens $\{Z^{(1)}, \dots, Z^{(t-1)}\}$. However, this approach is only equivalent to the exact model of the joint distribution $\mathbb{P}_Z$ if the codes of each codebook are mutually independent, conditionally to the past codes. Using the concepts introduced in 2.2, this means the family $\{Z_1^{(t)}, \dots Z_K^{(t)}\}$ should be independent, conditionally to $\{Z^{(1)}, \dots, Z^{(t-1)}\}$. As t increases, errors due to statistical dependence between codes may compound and cause the model to diverge from the true distribution. However, this method preserves the original codec frame rate, significantly accelerating training and inference.

Several alternative decoding strategies have been introduced: (Wang et al., 2023) propose to fully model the distribution of the first codebook, then to learn the factorized distribution over the remaining codebooks, while (Borsos et al., 2023; Agostinelli et al., 2023) model the first four codebooks with a first decoder, then the remaining eight codebooks with a second decoder. (Kharitonov et al., 2022) introduce a delay between codebooks for multi-stream language modeling, as an alternative to simply modelling all codebooks in parallel. This was used for audio and music generation in (Kreuk et al., 2023) and (Copet et al., 2023), respectively.

We propose instead to address the issue of statistical dependence between codes, so that we can reduce the modelling error but keep the inference time low when modelling the factorized distribution. This is the objective of the next section, where we present our independence promoting loss.

3. Method

We introduce here our proposed objective loss for promoting independence between codebooks. Using the maximum mean discrepancy framework presented in Section 2.2, we choose a reproducible kernel Hilbert space $\mathbb{H}$ equipped with a kernel $k(\cdot, \cdot)$. We do not operate in a variational framework, and consequently do not posit assumptions as to how the codes are distributed in the latent space. Therefore, we need to work with empirical estimators. An unbiased empirical estimator for the MMD lower-bound between samples $\{Z_i\}_{i=1}^B$ and $\{\bar{Z}_i\}_{i=1}^B$ is obtained from (6):

$$\begin{aligned}\mathrm{MMD}_{\mathbb{H}}(\mathbb{P}_Z||\mathbb{P}_{\bar{Z}}) = &\frac{1}{B(B-1)}\sum_{i=1}^{B}\sum_{j\neq i}k(Z_i, Z_j)\\ &+\frac{1}{B(B-1)}\sum_{j=1}^{B}\sum_{j\neq i}k(\bar{Z}_i, \bar{Z}_j)\\ &-\frac{2}{B^2}\sum_{i=1}^{B}\sum_{j=1}^{B}k(Z_i, \bar{Z}_j),\end{aligned} \tag{7}$$

where B is the sample size and i, j are indexes of samples in the batch.

Given a batch of samples $\{Z_i\}_{i=1}^B$ of the joint distribution $\mathbb{P}_Z$ obtained via encoding and quantization, we use the same batch shuffling strategy as (Brakel & Bengio, 2017) to obtain samples $\{\bar{Z}_i\}_{i=1}^B$ of the factorized distribution $\mathbb{P}_{\bar{Z}}$. For each codebook, we randomly shuffle the corresponding codes along the batch dimension, which was shown to effectively approximate samples of the factorized distribution $\mathbb{P}_{\bar{Z}}$ for sufficiently large sample sizes B. As explained further in the experiments section, we choose the sample size to be as large as possible to reduce both the variance of the empirical $\mathrm{MMD}_{\mathbb{H}}$ estimator and the reshuffling algorithm. The independence loss $\mathcal{L}_{\text{inde}}$ is then obtained by computing the empirical $\mathrm{MMD}_{\mathbb{H}}$ estimator between samples from the joint and approximate factorized distributions, as summarized in Algorithm 1. Note that by promoting independence between codeboks through optimization of $\mathrm{MMD}_{\mathbb{H}}$, we actually achieve more than the weaker *conditional* independence required by our decoding strategies to obtain exact modeling. Designing a conditional independence objective is not explored here.

This version of the proposed auxiliary loss promotes independence between the codes corresponding to encoded frames with similar frame index. This is optimal when adopting a parallel decoding strategy, effectively modelling the factorized distribution $\mathbb{P}_{\bar{Z}}$. We propose to extend our independence-promoting by applying the "delay" strategy proposed in (Kharitonov et al., 2022) to the codes before computing the $\mathrm{MMD}_{\mathbb{H}}$ estimator, effectively promoting independence between time-delayed codes $\{Z_k^{(\cdot -k+1)}\}_{k=1}^K$, as this will be our token decoding strategy for language modelling. The same could be done for other decoding strategies such as e.g. Vall-E (Wang et al., 2023). A diagram of the whole framework is displayed in Figure 1.

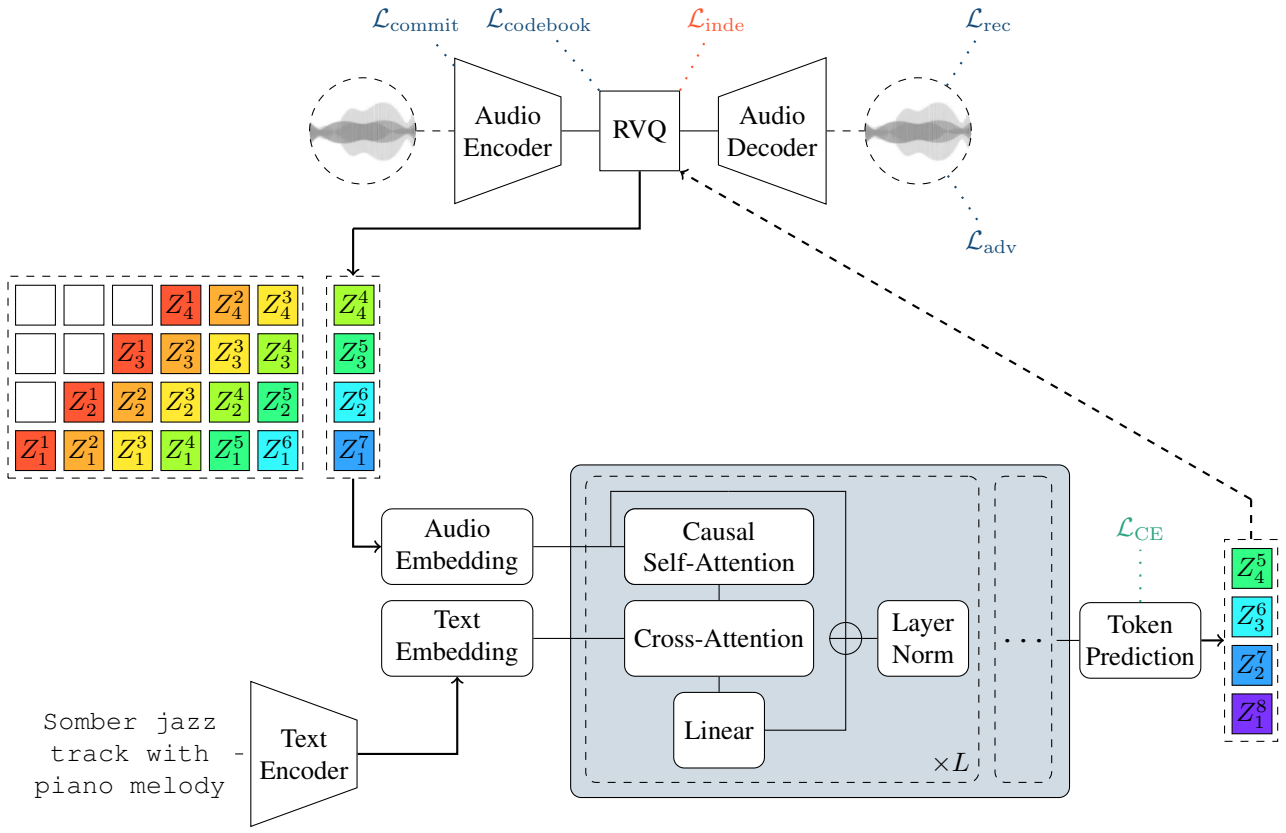

Figure 1. MusicGen framework. The EnCodec audio auto-encoder (top) encodes the waveform and audio tokens (middle) are obtained by discretizing the encoded audio with the RVQ multi-stage quantizer. The resulting audio tokens are then passed along text embeddings (bottom-left) to a Transformer-style language model with L layers (bottom-right). The language model auto-regressively estimates the next token (right) according to the "delay" decoding strategy (Kharitonov et al., 2022). At the time step $t = 7$, our proposed method MusicGen-MMD regularizes the EnCodec bottleneck with the loss $\mathcal{L}_{\text{inde}}$, thereby promoting independence between the delayed codes $\{Z_1^7, Z_2^6, Z_3^5, Z_4^4\}$ produced by RVQ.

Algorithm 1 MMD Optimization

Input: Training macro-batch X `%B,L`
Encode $X_e = \mathcal{E}_\theta(X)$ `%B,T,D`
Quantize $Z = \mathcal{Q}(X_e)$ `%B,K,T,N`
Optional: Apply "delay" $Z^{(t)}_{\cdot,k} = Z^{(t-k+1)}_{\cdot,k}$
Group time with batch axes $Z_{\cdot,k} \leftarrow Z_{\cdot,k,\cdot}$ `%B*T,K,N`
for codebook index $k \in \{1, \ldots, K\}$ **do**
  Sample permutation $\pi \sim \mathcal{U}(\mathcal{S}_{BT})$
  Shuffle batch axis $\{\bar{Z}_{i,k}\}_{i=1}^{BT} = \{Z_{\pi(i),k}\}_{i=1}^{BT}$
end for
Compute independence loss (7) $\mathcal{L}_{\text{inde}} = \text{MMD}(\mathbb{P}_Z || \mathbb{P}_{\bar{Z}})$

4. Experiments

4.1. Models and Hyperparameters

Auto-encoder: We use the 32kHz configuration of EnCodec (Défossez et al., 2023) as our audio tokenizer. EnCodec is a convolutional encoder-decoder model producing embeddings at 50 Hz for input waveforms sampled at 32 kHz. Each embedding is modeled by a RVQ scheme using 4 codebooks with $2^{11} = 2048$ entries each, which leads to an effective bitrate of 2.2kB.s^{-1}. The model is trained with a reconstruction loss ($\mathcal{L}_{\text{rec}}$) using a combination of L^1 and L^2 losses on the mel-spectrogram using multiple time resolutions (MSSpec), and a L^1 loss on the time signal. A multi-scale STFT discriminator is used to increase the reconstruction quality through adversarial training ($\mathcal{L}_{\text{adv}}$), and a feature matching loss is added for the training of the generator (Kumar et al., 2019). The quantizer is trained with the codebook loss ($\mathcal{L}_{\text{codebook}}$), and the encoder is additionally trained with a commitment loss pulling the encoder outputs closer to the learnt embeddings ($\mathcal{L}_{\text{commit}}$). Models are trained for 600k steps on 8 V100 GPUs with the Adam optimizer, using $\beta_1 = 0.5$, $\beta_2 = 0.9$, a learning rate of $3 \cdot 10^{-4}$, a batch size of 64 and segments of 1 second cropped at random in audio sequences.

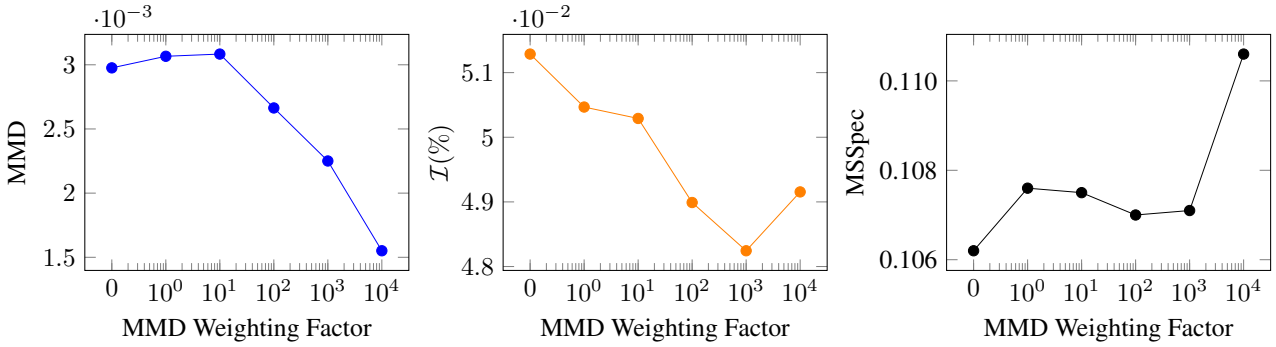

Figure 2. MMD, total correlation of EnCodec codes and MSSpec loss computed on our internal set. MSSpec is the combination of $L1$ and $L2$ losses on the multi-resolution mel-spectrogram, used for reconstruction in EnCodec. The horizontal axis shows the weighting factor used for the MMD loss $\mathcal{L}_{\text{inde}}$. The total correlation $\mathcal{I}$ is computed on the whole 250k-samples training set for minimal bias in the histogram approximation. It is computed between two codebooks taken at random, averaged over five codebook couples, and expressed as a ratio to the entropy of the joint distribution (in %).

Language Model: We train the same Transformer model as MusicGen-small (Copet et al., 2023), consisting of several Transformer-style layers for a total number of 300M parameters. Each layer comprises a causal self-attention module, a module computing cross-attention between the current signal and the conditioning text representation, a fully-connected block with ReLU, and a residual connection skipping from the layer's input. Sinusoidal positional encoding is used to embed the current time step (Vaswani et al., 2017). The decoding strategy for all models is the "delay" pattern (Kharitonov et al., 2022). The model is trained on cross-entropy ($\mathcal{L}_{\text{CE}}$) for 1M steps on 32 V100 GPUs with the AdamW optimizer, using $\beta_1 = 0.9$, $\beta_2 = 0.95$, a batch size of 192, and audio sequences of 30 seconds. We use a cosine learning rate schedule with a 4000-steps warmup. Exponential moving average with a decay of 0.99 is used to recursively smooth model weights. Top-250 sampling is used with a temperature of 1 during inference (Fan et al., 2018). The EnCodec audio codec and the text encoder are frozen during the training of the language model.

Text Conditioning: We use the T5 Transformed-based text encoder (Raffel et al., 2023). Metadata such as key, tempo or instrumentation are concatenated to the text description. We implement classifier-free guidance when sampling from the model's logits, as in (Kreuk et al., 2023). Therefore, we drop the conditioning signal with a probability of 0.2 during training, and at inference we use a guidance strength of 3.0.

Independence Loss: We use a weight of 10^3 for the independence loss $\mathcal{L}_{\text{inde}}$, computed in a separate backward. All the other losses are optimized as in (Défossez et al., 2023). We choose this value empirically by selecting the largest weighting factor that did not degrade the traditional EnCodec loss, as detailed in the ablation study in Section 5.1. The RKHS $\mathbb{H}$ is equipped with the multi-scale Gaussian kernel $k(x, y) = \sum_{\sigma_i} e^{-||x-y||^2/2\sigma_i^2}$ with radii $\sigma_i \in \{0.1, 1, 5, 10, 20, 50\}$. Therefore, it satisfies $\text{MMD}_{\mathbb{H}}(\mathbb{P}_Z||\mathbb{P}_{\bar{Z}}) = 0 \iff \mathbb{P}_Z = \mathbb{P}_{\bar{Z}}$ (see Section 2.2). We let the kernel functions fixed throughout training, although optimizing the standard deviations σ could lead to a better lower-bound of the true MMD in (6). This is because the distributions $\mathbb{P}_Z$ and $\mathbb{P}_{\bar{Z}}$ are being learnt as we compute the $\text{MMD}_{\mathbb{H}}$ estimator, therefore measuring the optimality of the chosen kernel $k(\cdot, \cdot)$ (or equivalently RKHS $\mathbb{H}$) is intrinsically hard. Furthermore, this would require a significant amount of energy spent in extensive grid searches, which we believe was not the focus of this study. We further justify the choice of the multi-scale Gaussian kernel in Section 5.4.

Unless mentioned otherwise, we use the decoding strategy adaptation proposed in Section 3 for the "delay" pattern (Kharitonov et al., 2022). We noticed in our experiments that although the estimator (7) is unbiased, a high batch size is required to reduce the variance of the estimator and properly optimize the objective $\mathcal{L}_{\text{inde}}$. We maximize the macro-batch size B by accumulating 32 batches, which results in $B = \text{batches} \times \text{batchsize} \times \tilde{T}/\text{gpus} = 1280$ samples per GPU). We make these samples fit on a V100 GPU by using gradient checkpointing during encoding to compute the independence loss in a separate computational graph, which significantly reduces the amount of GPU memory used, at a minor increase in training time.

4.2. Datasets

We use 20K hours of licensed music to train both EnCodec and the language model. The training dataset is composed of an internal dataset of 10K high-quality music tracks, and the ShutterStock and Pond5 music data collections[2], respectively consisting of 25K and 365K music tracks. All datasets comprise full-length music samples recorded at 32 kHz, ac-

[2]www.shutterstock.com/music www.pond5.com

companied by metadata including a textual description and supplementary details such as genre, key, tempo, etc. For comparison of the proposed method to the baselines, we employ the MusicCaps benchmark (Agostinelli et al., 2023) as our primary evaluation dataset. MusicCaps comprises 5.5K samples, each lasting ten seconds and curated by expert musicians. We resample all samples to 16kHz for fairness. For ablation studies, we rely on a held-out internal evaluation set featuring 528 music tracks.

4.3. Evaluation Metrics

We conduct a comprehensive evaluation using both objective and subjective metrics. Objective functions include the Fréchet Audio Distance (FAD) (Kilgour et al., 2019) computed as the distance between Gaussian distributions fitted on DNN-obtained embeddings of the real and generated samples. As highlighted in (Gui et al., 2024), using FAD can lead to wrong interpretations if using irrelevant embeddings. We therefore use various embeddings such as CLAP-Laion (contrastive learning audio pretraining), MERT-4 (acoustic music understanding) and VGGish (audio feature classification)[3]. To complement this, akin to (Yang et al., 2023b), we calculate the KL-Divergence between the outputs of the Patch-Out-Transformer[4] audio classifier (Koutini et al., 2022), utilizing the original and generated audio as inputs. These metrics deliver insights into complementary aspects of the generated audio, namely quality, fidelity and high-level semantics.

For subjective evaluation, we conducted a MUSHRA-style mean opinion score (MOS) test, where 11 annotators were each asked to rate 12 samples each with a single number between 0 and 100 representing the overall music quality, including audio quality as well as consistency and likelihood of the harmonic, melodic and rhythmic structure. The ground-truth reference was given (and hidden among the samples for rating) as an anchor representing a music track with maximum music quality. The files rated by the annotators were randomly drawn from the MusicCaps dataset, normalized at -14dB LUFS(ITU-R, 2017). The text description was not shown during the test. See Appendix F for more details. We also run a second subjective evaluation with annotators recruited via Amazon Mechanical Turk: results and methodology are reported in Appendix G.

4.4. Baselines

We compare our proposed method trained for music generation to the original MusicGen model without independence loss (Copet et al., 2023), as well as other state-of-the-art latent diffusion baselines such as the text-to-music version of AudioLDM2 (Liu et al., 2023b)[5] (denoted as AudioLDM2-Music in the following) , its predecessor AudioLDM (Liu et al., 2023a)[6], and Mustango (Melechovsky et al., 2023)[7]. For completeness we also include other language modelling baselines such as MusicLM (Agostinelli et al., 2023), Noise2Music (Huang et al., 2023) and the recent audio fondational model UniAudio (Yang et al., 2023a). For these however, we were not able to evaluate these baselines as the public implementation was not made available for the given text-to-music generation task, and therefore reported results from the original papers directly.

5. Results

We introduce our results section by running an analysis of the proposed independence-proxy loss with respect to the weighting factor used for optimization, and investigate its correlation with total correlation of the codes. We follow by reporting objective and subjective metrics for music generation on the standard MusicCaps benchmark. Then, we proceed with an ablation study to show the efficiency of integrating the decoding strategy for MMD loss optimization. We also test the generalizibility of our method by applying it to a different state-of-the-art audio codec, namely RVQ-GAN (Kumar et al., 2024), and we analyse the resulting performance in appendix B. Finally, we conduct ablation studies with respect to other quantization schemes: results are reported in appendices B,C and D.

5.1. MMD as an Independence-promoting Loss

We show in Figure 2 the MMD, total correlation and MSSpec loss values for EnCodec codes (which are later used as tokens in our language model). We show our grid search with respect to the scaling factor for the MMD loss. We use our whole 250k-samples internal set for minimal bias in histogram approximation. The total correlation $\mathcal{I}$ is computed between two codebooks taken at random, averaged over five codebook couples, and expressed as a ratio to the entropy of the joint distribution (in %). We first observe that MMD overall correlates with the total correlation, which shows that our proposed loss is a reasonable independence proxy. Except for the large weighting factor of 10^4, the MMD loss and total correlation diminish monotonously with respect to the weighting factor used for optimization, which qualifies the proposed criterion as a valid objective loss. The MSSpec reconstruction loss remains unaffected except when using a very large scaling factor of 10^4, for which the training seems perturbed, and where the total correlation does not seem to correlate with MMD anymore. We choose a factor of 10^3 as it allows a maximal total corre-

[3]We compute all these scores using the official repository https://github.com/microsoft/fadtk associated to (Gui et al., 2024).

[4]https://github.com/kkoutini/PaSST

[5]https://github.com/haoheliu/AudioLDM2

[6]https://github.com/haoheliu/AudioLDM

[7]https://github.com/AMAAI-Lab/mustango

Table 1. Text-to-music generation on MusicCaps. Asterisks* mean that we report figures from the related papers as the public implementation was not available for the given text-to-music generation task. Mustango was trained on an augmented version of MusicCaps, therefore we put it aside the other baselines. For the subjective metric (OVRL), mean and 95 % confidence intervals are showed. The samples presented were 10 second long sampled at 16kHz, which matches the training conditions of Mustango, AudioLDM and AudioLDM2-Music. In comparison MusicGen and MusicGen-MMD were trained on 30 second-long segments sampled at 32kHz.

Model	# params	$\text{FAD}_{clap-laion} \downarrow$	$\text{FAD}_{MERT-4} \downarrow$	$\text{FAD}_{vgg} \downarrow$	KL $\downarrow$	CLAP (%) $\uparrow$	OVRL. $\uparrow$
Ground-Truth		-	-	-	-	*38*	*97.95 ± 1.13*
Mustango	1.4 B	*0.07*	*1.65*	*1.56*	*0.71*	*37*	*49.26 ± 4.21*
MusicLM*	860 M	-	-	4.0	-	-	-
Noise2Music*	1.3 B	-	-	2.1	-	-	-
UniAudio*	1 B	-	-	3.65	1.87	-	-
AudioLDM	416 M	0.18	4.18	3.52	1.42	**35**	56.29 ± 4.35
AudioLDM2-Music	347 M	0.25	4.30	4.71	1.31	31	69.43 ± 3.42
MusicGen	300 M	0.16	1.57	3.60	1.22	31	62.54 ± 3.68
MusicGen-MMD (ours)	300 M	**0.14**	**1.45**	**2.98**	**1.18**	32	**74.75 ± 3.68**

Table 2. Text-to-music generation results on held-out test set. All models have 300M parameters.

MusicGen Configuration	$\text{FAD}_{vgg} \downarrow$	KL $\downarrow$	CLAP (%) $\uparrow$
Ground-truth	-	-	*38*
Delay (Copet et al., 2023)	0.95	**0.45**	37
Delay w/ MMD-Parallel	0.90	**0.45**	37
Delay w/ MMD (proposed)	**0.59**	0.46	37
Flatten	0.69	0.46	**39**

Table 3. MMD, total correlation and reconstruction losses of EnCodec-MMD with various kernels evaluated on our internal dataset. We used a weight of 1000 for the MMD loss, and adapted the weighting factors of the MMD loss so that the magnitudes of the losses stayed approximately consistent across kernels. The total correlation $\mathcal{I}$ is computed on the whole 250k-samples training set for minimal bias in the histogram approximation. It is calculated between two codebooks taken at random, averaged over five codebook couples, and expressed as a ratio to the entropy of the joint distribution (in %).

Method	$\mathcal{I}$ (%) $\downarrow$	MSMelSpec $\downarrow$
Multi-Scale Gaussian	$4.8 \cdot 10^{-2}$	**0.107**
Squared Inverse	$\mathbf{4.1} \cdot 10^{-2}$	0.127
Linear	$5.0 \cdot 10^{-2}$	0.114
Quadratic	$4.9 \cdot 10^{-2}$	0.118

lation reduction without hurting the reconstruction loss.

We show in Appendix B that our method is generalizable to other codecs, by applying MMD optimization to the latent space of RVQGAN (Kumar et al., 2024), which is a state-of-the-art audio codec based on EnCodec. Our results support that MMD optimization can also be used to promote the independence of RVQGAN codes, in a similar fashion to what have demonstrated here for EnCodec codes.

5.2. Text-to-Music Generation Benchmark

We show objective and subjective evaluation results for music generation on MusicCaps in Table 1. We observe that the objective metrics of Mustango are quite strong, as the model was trained on an augmented version of MusicCaps. Our method MusicGen-MMD improves objective metrics over our own baseline MusicGen, and obtains better objective metrics than AudioLDM, AudioLDM2-Music, MusicLM and UniAudio. Noise2Music still obtains a better FAD result, although with a much larger architecture (1.3 B). Furthermore, we could not reproduce the results nor run other metrics (such as FAD with other embeddings) as the implementation was not made publicly available. The subjective metric OVRL. obtained via the MUSHRA-style test indicates that our model MusicGen-MMD obtains the best performance, closely followed by AudioLDM2-Music. Then follow MusicGen, AudioLDM and finally Mustango.

5.3. Decoding Strategy Matching

We present the effect of integrating the language model decoding strategy to the MMD loss optimization. We train three models with the same language modeling configuration and the "delay" decoding stategy, but distinct EnCodec configurations: our baseline without MMD optimization (Delay), our proposed model using the "delay" decoding strategy for optimizing the MMD (Delay w/ MMD) and our proposed model where the MMD optimizes does not integrate the decoding strategy (Delay w/ MMD-Parallel). Finally we train a MusicGen model using the "flatten" decoding strategy where the codebooks are flattened such that a single code is predicted at each time step. This effectively models the joint distribution $\mathbb{P}_Z$ instead of the factorized distribution $\mathbb{P}_{\tilde{Z}}$. Results are computed on our held-out test set and reported in Table 2. Objective scores show that adapting the MMD optimization to the language modelling decoding strategy improves audio quality and fidelity, as our proposed method obtains a better FAD_{vgg} than the one where the

MMD criterion is not adapted to the language model decoding strategy. Our method even outperforms the MusicGen with "flatten" strategy on the FAD_{vgg} score, which indicates that training the language model to predict the joint distribution by flattening the codebooks does not yield optimal performance, which we posit is due to increased training difficulty. In addition, the original frame rate of EnCodec is preserved, whereas MusicGen with "flatten" decoding largely increases the inference time, by a factor equal to the number of codebooks K.

5.4. Kernel Function Ablation

We justify here how the choice of kernel function $k(\cdot, \cdot)$ impacts the reconstruction error of EnCodec and the total correlation of the codes.

First, the Gaussian kernel is a natural candidate as it is widely used in statistics and machine learning. Furthermore, we observed experimentally that using several standard deviations σ_i increases the numerical robustness of the MMD computation, as unadapted values might make the exponentials in the Gaussian kernel collapse to values where the numerical rounding errors degrade the estimation of the MMD. Using several σ_i therefore enables us to avoid this pitfall, as we can expect at least some values to produce reliable estimates.

We have conducted experiments with a variety of other kernels and provide the results in Table 3. The squared inverse kernel is defined here as $k(x, y) = (1 + (||x - y||^2)/\sigma^2)^{-1}$ with $\sigma = 12$, the linear kernel as $k(x, y) = x^T y$ and the quadratic kernel as $k(x, y) = (x^T y)^2$. We observe that the multi-scale Gaussian kernel achieves the most interesting trade-off, by obtaining the second lowest total correlation while outperforming all other kernel functions on reconstruction, thereby justifying its choice in our subsequent experiments.

6. Conclusion

We presented an independence-proxy loss for regularizing discrete latent representations used as tokens in music generation language models. We showed that the proposed method outperforms our baseline and other state-of-the-art music generation models, without adding parameters nor increasing the inference time compared to the baseline. We performed an analysis of the propose criterion, showing its correlation with total correlation of the codes and investigating the effects of adapting the criterion to the decoding strategy used in further language modelling. We also demonstrated that the proposed criterion can be easily plugged into other multi-stream codecs, and more generally we would argue that is is a reasonable independence optimization criterion for other applications than music generation.

Impact Statement

Large scale generative models boast high expression capabilities, which raises questions regarding ethics and societal consequences of their use. In particular, text-to-music generative models can constitute an unfair competition for musicians (and artists and creators in general). This is a societal issue that has not been solved yet and demands serious regulatory investigation. We try and make our research as open and accessible as possible, ensuring that the involved parties, both amateurs and professional, have equal access to the developed methods. Another potential bias towards individuals resides in the large proportion of Western music (and in particular pop instrumental and electronic music) of the data used to train our model, which resents a lack of diversity. However, the somewhat reasonable size of the model presented in this paper and the low number of auto-regressive steps used for inference should encourage reproducibility of our method for new data sources.

References

Agostinelli, A., Denk, T. I., Borsos, Z., Engel, J., Verzetti, M., Caillon, A., Huang, Q., Jansen, A., Roberts, A., Tagliasacchi, M., Sharifi, M., Zeghidour, N., and Frank, C. MusicLM: Generating music from text. *arXiv preprint arXiv:2301.11325*, 2023.

Arjovsky, M., Chintala, S., and Bottou, L. Wassertein generative adversarial networks. *Proc. Int. Conf. Machine Learning*, 2017.

Belghazi, M. I., Baratin, A., Rajeswar, S., Ozair, S., Bengio, Y., Courville, A., and Hjelm, R. D. MINE: Mutual Information Neural Estimation. *Proc. Int. Conf. Machine Learning*, 2018.

Borsos, Z., Marinier, R., Vincent, D., Kharitonov, E., Pietquin, O., Sharifi, M., Roblek, D., Teboul, O., Grangier, D., Tagliasacchi, M., and Zeghidour, N. AudioLM: a language modeling approach to audio generation. *CoRR*, 2023.

Brakel, P. and Bengio, Y. Learning independent features with adversarial nets for non-linear ICA. *Proc. Int. Conf. Machine Learning*, 2017.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. *Proc. Neural Inf. Process. Syst.*, 2020.

Burgess, C. P., Higgins, I., Pal, A., Matthey, L., Watters, N., Desjardins, G., and Lerchner, A. Understanding disentangling in β-VAE. *Proc. Neural Inf. Process. Syst.*, 2017.

Copet, J., Kreuk, F., Gat, I., Remez, T., Kant, D., Synnaeve, G., Adi, Y., and Défossez, A. Simple and controllable music generation. *Proc. Neural Inf. Process. Syst.*, 2023.

Défossez, A., Copet, J., Synnaeve, G., and Adi, Y. High fidelity neural audio compression. *Transactions on Machine Learning Research*, 2023.

Dhariwal, P., Jun, H., Payne, C., Kim, J. W., Radford, A., and Sutskever, I. Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341*, 2020.

Fan, A., Lewis, M., and Dauphin, Y. Hierarchical neural story generation. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, 2018.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, F., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial networks. *Proc. Neural Inf. Process. Syst.*, 2014.

Gray, R. M. Vector quantization. *IEEE ASSP Magazine*, 1984.

Gretton, A., Bordwardt, K., Rasch, M., Schoelopf, B., and Smola, A. A kernel two-sample test. *Journal of Machine Learning Research*, 2012.

Gui, A., Gamper, H., Braun, S., and Emmanouilidou, D. Adapting frechet audio distance for generative music evaluation. In *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2024. doi: 10.1109/ICASSP48485.2024.10446663.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. β-vae: Learning basic visual concepts with a constrained variational framework. *Proc. Int. Conf. Learning Repr.*, 2017.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Proc. Neural Inf. Process. Syst.*, 2020.

Huang, Q., Park, D. S., Wang, T., Denk, T. I., Ly, A., Chen, N., Zhang, Z., Zhang, Z., Yu, J., Frank, C., Engel, J., Le, Q. V., Chan, W., Chen, Z., and Han, W. Noise2Music: Text-conditioned music generation with diffusion models. *arXiv preprint arXiv:2302.03917*, 2023.

Huszar, F. An alternative update rule for generative adversarial networks. *Blogpost*, 2016.

Hyvarinen, A., Khemakhem, I., and Morioka, H. Nonlinear Independent Component Analysis for Principled Disentanglement in Unsupervised Deep Learning. *Patterns*, 2023.

ITU-R. Algorithms to measure audio programme loudness and true-peak audio level. 2017.

Ju, Z., Wang, Y., Shen, K., Tan, X., Xin, D., Yang, D., Liu, Y., Leng, Y., Song, K., Tang, S., Wu, Z., Qin, T., Li, X.-Y., Ye, W., Zhang, S., Bian, J., He, L., Li, J., and Zhao, S. Naturalspeech 3: Zero-shot speech synthesis with factorized codec and diffusion models. In *arXiv preprint arXiv:2403.03100*, 2024.

Juang, B.-H. and Gray, A. Multiple stage vector quantization for speech coding. *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 1982.

Kharitonov, E., Lee, A., Polyak, A., Adi, Y., Copet, J., Lakhotia, K., Nguyen, T.-A., Rivière, M., Mohamed, A., Dupoux, E., and Hsu, W.-N. Text-free prosody-aware generative spoken language modeling. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 2022.

Kilgour, K., Zuluaga, M., Roblek, D., and Sharifi, M. Fréchet audio distance: A metric for evaluating music enhancement algorithms. *INTERSPEECH*, 2019.

Kingma, D. and Welling, M. Auto-encoding variational bayes. *Proc. Int. Conf. Learning Repr.*, 2014.

Kong, J., Kim, J., and Bae, J. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *Proc. Neural Inf. Process. Syst.*, 2020.

Kong, Z., Ping, W., Huang, J., Zhao, K., and Catanzaro, B. Diffwave: A versatile diffusion model for audio synthesis. *Proc. Int. Conf. Learning Repr.*, 2021.

Koutini, K., Schlüter, J., Eghbal-zadeh, H., and Widmer, G. Efficient training of audio transformers with patchout. *Proc. Interspeech*, 2022.

Kreuk, F., Synnaeve, G., Polyak, A., Singer, U., Défossez, A., Copet, J., Parikh, D., Taigman, Y., and Adi, Y. Audiogen: Textually guided audio generation. *Proc. Int. Conf. Learning Repr.*, 2023.

Kumar, K., Kumar, R., de Boissiere, T., Gestin, L., Teoh, W. Z., Sotelo, J., de Brebisson, A., Bengio, Y., and Courville, A. Melgan: Generative adversarial networks for conditional waveform synthesis. *Proc. Neural Inf. Process. Syst.*, 2019.

Kumar, R., Seetharaman, P., Luebs, A., Kumar, I., and Kumar, K. High-fidelity audio compression with improved rvqgan, 2024.

Li, H., YU, S., and Principe, J. Deep deterministic independent component analysis for hyperspectral unmixing. *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2023.

Liu, H., Chen, Z., Yuan, Y., Mei, X., Liu, X., Mandic, D., Wang, W., and Plumbley, M. D. AudioLDM: Text-to-audio generation with latent diffusion models. *Proc. Int. Conf. Machine Learning*, 2023a.

Liu, H., Tian, Q., Yuan, Y., Liu, X., Mei, X., Kong, Q., Wang, Y., Wang, W., Wang, Y., and Plumbley, M. D. AudioLDM 2: Learning holistic audio generation with self-supervised pretraining. *arXiv preprint arXiv:2308.05734*, 2023b.

Melechovsky, J., Guo, Z., Ghosal, D., Majumder, N., Herremans, D., and Poria, S. Mustango: Toward controllable text-to-music generation. *arXiv preprint arXiv:2311.08355*, 2023.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. *Technical Report*, 2019.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 2023.

Ribeiro, F., Florêncio, D., Zhang, C., and Seltzer, M. Crowdmos: An approach for crowdsourcing mean opinion score studies. *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2011.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, 2022.

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. *Proc. Neural Inf. Process. Syst.*, 2019.

van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. Wavenet: A generative model for raw audio. 2016.

Vasuki, A. and Vanathi, P. A review of vector quantization techniques. *IEEE Potentials*, 2006.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. *Proc. Neural Inf. Process. Syst.*, 2017.

Villani, C. Optimal transport: Old and new. *Grundlehren der mathematischen Wissenschaften*, 2009.

Wang, C., Chen, S., Wu, Y., Zhang, Z., Zhou, L., Liu, S., Chen, Z., Liu, Y., Wang, H., Li, J., He, L., Zhao, S., and Wei, F. Neural codec language models are zero-shot text to speech synthesizers. *arXiv preprint arXiv:2301.02111*, 2023.

Yang, D., Tian, J., Tan, X., Huang, R., Liu, S., Chang, X., Shi, J., Zhao, S., Bian, J., Wu, X., Zhao, Z., Watanabe, S., and Meng, H. Uniaudio: An audio foundation model toward universal audio generation. *arXiv preprint arXiv:2310.00704*, 2023a.

Yang, D., Yu, J., Wang, H., Wang, W., Weng, C., Zou, Y., and Yu, D. Diffsound: Discrete diffusion model for text-to-sound generation. *IEEE/ACM Trans. Audio Speech Lang. Process.*, 2023b.

Yu, S., Alesiani, F., Yu, X., Jenssen, R., and Principe, J. C. Measuring Dependence with Matrix-based Entropy Functional. *AAAI*, 2021.

Zeghidour, N., Luebs, A., Omran, A., Skoglund, J., and Tagliasacchi, M. SoundStream: An end-to-end neural audio codec. *arXiv preprint arXiv:2107.03312*, 2021.

Zhang, X., Zhang, D., Li, S., Zhou, Y., and Qiu, X. Speechtokenizer: Unified speech tokenizer for speech large language models. In *Proc. Int. Conf. Learning Repr.*, 2024.

A. Proof of Kernel Formulation of MMD

This is the proof to (6) and mostly uses material from (Gretton et al., 2012). First, the notion of feature mapping can be extended to the *mean embedding* of a probability distribution (Gretton et al., 2012). Given a probability distribution $\mathbb{P}_X$ we define its mean embedding $\mu_{\mathbb{P}_X} := \mathbb{E}_{X\sim\mathbb{P}_X}[\phi(X)] \in \mathbb{H}$ such that:

$$\forall\, h \in \mathbb{H} : \mathbb{E}_{X\sim\mathbb{P}_X}[h(X)] = \langle h, \mu_{\mathbb{P}_X}\rangle_{\mathbb{H}} \tag{8}$$

If h is taken to be in a RKHS $\mathbb{H}$, the obtained MMD estimate is actually a lower-bound of the true MMD:

$$\begin{aligned}\mathrm{MMD}(\mathbb{P}_Z||\mathbb{P}_{\bar{Z}}) &= \sup_{h,\|h\|\leq 1} \mathbb{E}_{\mathbb{P}_Z}[h(\bar{Z})] - \mathbb{E}_{\mathbb{P}_{\bar{Z}}}[T(\bar{Z})] \\ &\geq \underbrace{\sup_{h\in\mathbb{H},\|h\|\leq 1} \mathbb{E}_{\mathbb{P}_Z}[h(\bar{Z})] - \mathbb{E}_{\mathbb{P}_{\bar{Z}}}[T(\bar{Z})]}_{\mathrm{MMD}_{\mathbb{H}}(\mathbb{P}_Z||\mathbb{P}_{\bar{Z}})}.\end{aligned}$$

Using (8) in (5) and the properties of $\mathbb{H}$, we can then compute the MMD between $\mathbb{P}_Z$ and $\mathbb{P}_{\bar{Z}}$ taking the supremum over the unit ball of $\mathbb{H}$ as:

$$\begin{aligned}\mathrm{MMD}_{\mathbb{H}}(\mathbb{P}_Z||\mathbb{P}_{\bar{Z}}) &= \sup_{h\in\mathbb{H},\|h\|\leq 1} \mathbb{E}_{\mathbb{P}_Z}[h(\bar{Z})] - \mathbb{E}_{\mathbb{P}_{\bar{Z}}}[T(\bar{Z})] \\ &= \sup_{h\in\mathbb{H},\|h\|\leq 1} \langle h, \mu_{\mathbb{P}_Z} - \mu_{\mathbb{P}_{\bar{Z}}}\rangle \\ &= \left\|\mu_{\mathbb{P}_Z} - \mu_{\mathbb{P}_{\bar{Z}}}\right\|_{\mathbb{H}} \\ &= \langle \mu_{\mathbb{P}_Z}, \mu_{\mathbb{P}_Z}\rangle - 2\langle \mu_{\mathbb{P}_Z}, \mu_{\mathbb{P}_{\bar{Z}}}\rangle + \langle \mu_{\mathbb{P}_{\bar{Z}}}, \mu_{\mathbb{P}_{\bar{Z}}}\rangle,\end{aligned}$$

where we use the 1-Lipschitz property of h in the third line. We can then use the definition of the mean embedding to obtain:

$$\begin{aligned}\mathrm{MMD}_{\mathbb{H}}(\mathbb{P}_Z||\mathbb{P}_{\bar{Z}}) = &\ \mathbb{E}_{Z_1\sim\mathbb{P}_Z}\mathbb{E}_{Z_2\sim\mathbb{P}_Z}\langle\phi(Z_1),\phi(Z_2)\rangle \\ &+ \mathbb{E}_{\bar{Z}_1\sim\mathbb{P}_{\bar{Z}}}\mathbb{E}_{\bar{Z}_2\sim\mathbb{P}_{\bar{Z}}}\langle\phi(\bar{Z}_1),\phi(\bar{Z}_2)\rangle \\ &- 2\mathbb{E}_{Z_1\sim\mathbb{P}_Z}\mathbb{E}_{\bar{Z}_2\sim\mathbb{P}_{\bar{Z}}}\langle\phi(Z_1),\phi(\bar{Z}_2)\rangle.\end{aligned}$$

Finally, using the kernel definition in $\mathbb{H}$:

$$\begin{aligned}\mathrm{MMD}_{\mathbb{H}}(\mathbb{P}_Z||\mathbb{P}_{\bar{Z}}) = &\ \mathbb{E}_{Z_1\sim\mathbb{P}_Z}\mathbb{E}_{Z_2\sim\mathbb{P}_Z}k(Z_1,Z_2) \\ &+ \mathbb{E}_{\bar{Z}_1\sim\mathbb{P}_{\bar{Z}}}\mathbb{E}_{\bar{Z}_2\sim\mathbb{P}_{\bar{Z}}}k(\bar{Z}_1,\bar{Z}_2) \\ &- 2\mathbb{E}_{Z_1\sim\mathbb{P}_Z}\mathbb{E}_{\bar{Z}_2\sim\mathbb{P}_{\bar{Z}}}k(Z_1,\bar{Z}_2).\end{aligned}$$

B. MMD Optimization on RVQGAN Codes

We apply here our MMD optimization method on RVQGAN (Kumar et al., 2024), a state-of-the-art codec based on EnCodec. RVQGAN improves upon EnCodec by using lower-dimensional embeddings in the RVQ codebooks, thereby increasing codebook utilization. The authors also propose a new multi-scale STFT discriminator and various other techniques to increase the quality at lower-bitrate regimes. Our aim here is to demonstrate that our independence-promoting criterion based on MMD optimization is generalizable to other codecs. We employ the same setup as in our main experiments, and simply use RVQGAN in place of EnCodec, keeping the number of codebooks and the total bandwidth identical. We show the MMD loss, mutual information of RVQGAN codes and reconstruction losses in Figure 3. We observe the similar trend compared to our method applied to EnCodec, with an even stronger correlation between the scale of the MMD loss and the mutual information, which implies that MMD optimization of the RVQGAN latent space also correlates with a more independence of the RVQGAN codes.

C. MMD Optimization with Different Quantization Schemes

Product vector quantization (PVQ) is another multistage quantization method, where the input vector dimensions are split across C groups and each group of dimensions is encoded by a codebook with dimensionality N/C. Although this scheme is typically non-hierarchical, since no priority is given to any particular codebook, a hierarchy can be introduced through hierarchical dropout (PVQ-dropout). This means sampling a natural number $k \sim \mathcal{U}(\{1,\ldots,K\})$ and using only the *first* k codebooks for encoding (and putting the other codes to 0 before decoding). This quantizer dropout technique is also used in the RVQ-based SoundStream codec (Zeghidour et al., 2021), however with a different intent: it allows the resulting codec to function at various bitrates without further adaptation at training time.

We employ here a similar setup as in Section 5.1. We show in Table 4 the MMD and total correlation values for EnCodec codes (which are later used as tokens in our language model), with the chosen scale factor of 10^3. We use our whole 250k-samples internal set for minimal bias in histogram approximation. The total correlation $\mathcal{I}$ is computed between two codebooks taken at random, averaged over five codebook couples, and expressed as a ratio to the entropy of the joint distribution (in %). We observe that residual quantization introduces more dependence between codes compared to product quantization, although both induce a hierarchical structure in the codes space, which accounts for their high coding efficiency. We also observe that our proposed MMD loss is able to curb both the MMD and total correlation of the PVQ w/ dropout codes, highlighting its versitality.

D. Effect of Hierarchy in Quantized Audio Space

We investigate here the performance of language models as a function of the quantization scheme used. We use three different quantizers for EnCodec: RVQ, which is our default

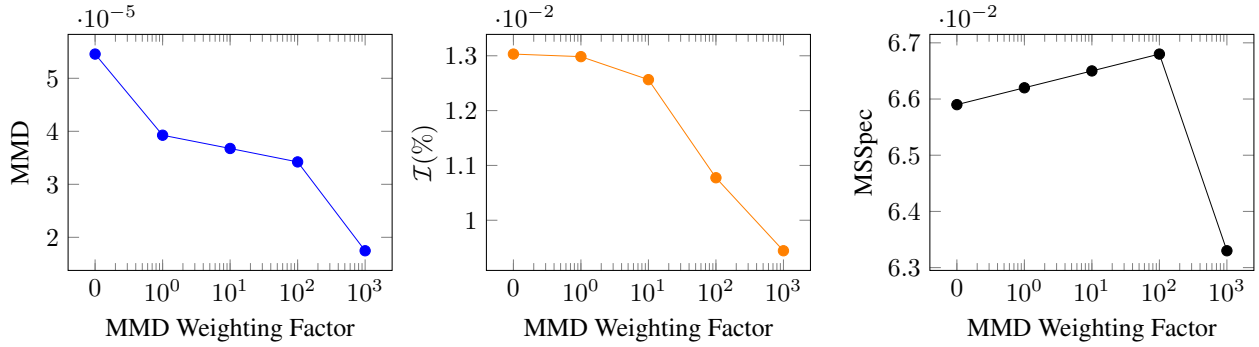

Figure 3. MMD, Mutual Information of RVQGAN (Kumar et al., 2024) codes and MSSpec loss computed on our internal set. MSSpec is the combination of $L1$ and $L2$ losses on the multi-resolution mel-spectrogram, used for reconstruction in EnCodec. The horizontal axis shows the weighting factor used for the MMD loss $\mathcal{L}_{\text{inde}}$. The total correlation $\mathcal{I}$ is computed on the whole 250k-samples training set for minimal bias in the histogram approximation. It is computed between two codebooks taken at random, averaged over five codebook couples, and expressed as a ratio to the entropy of the joint distribution (in %). We removed the data point for the MMD weight of 10^4 as the experiment diverged.

Table 4. MMD and total correlation of EnCodec codes. Results computed on complete 250k-samples internal set.

EnCodec Quantizer	MMD↓	$\mathcal{I}$ (%) ↓
RVQ	$9.9 \cdot 10^{-4}$	$5.1 \cdot 10^{-2}$
RVQ w/ MMD	$9.9 \cdot 10^{-5}$	$4.8 \cdot 10^{-2}$
PVQ w/ dropout	$3.7 \cdot 10^{-5}$	$3.8 \cdot 10^{-2}$
PVQ w/ dropout + MMD	$4.5 \cdot 10^{-7}$	$3.0 \cdot 10^{-2}$

quantizer, PVQ and PVQ-dropout. As explained in C, introducing a codebook dropout mechanism in PVQ naturally induces a hierarchical structure, as EnCodec will more regularly rely on the first few codebooks to reconstruct the audio. By looking at the contributions of individual codebooks (not shown here), we can observe a similar hierarchical structure for PVQ-dropout and RVQ, and no hierarchy in PVQ codes. We subsequently trained three language models with their respective EnCodec configurations (RVQ, PVQ, PVQ w/ dropout) and the same language model configuration. Objective results on our held-out test set are reported in Table 5. We observe that the model using PVQ has low objective scores, while that using PVQ w/ dropout obtains much better objective scores at language modeling, somewhat close yet still inferior to the RVQ-equipped model, which seems to be the best strategy here and demonstrates the high coding efficiency of residual vector quantization. This seems to indicate that hierarchical structure in the token space leads to better language modeling performance, which we posit is due to the language models being able to rely on its first few codebooks in case its modeling capacity it too limited. On the other hand, as we indicated in the main paper, promoting independence between codes for exact modeling of the codebook distributions is also theoretically motivated and experimentally demonstrated. This means there is potentially a trade-off to seek between hierarchy and independence in the codes space. The first is obtained via structural properties of the used quantizer e.g. residual quantization or dropout, and the second can be tuned via independence optimization as proposed in this paper. We argue that the complimentary nature of these solutions allows for a control over this trade-off for optimal audio generation performance.

Table 5. Text-to-Music generation of MusicGen with various quantization schemes for EnCodec tokenizer. Results are shown on the held-out test set. All models have 300M parameters.

EnCodec Quantizer	FAD_{vgg} ↓	KL ↓	CLAP (%) ↑
RVQ	**0.97**	**0.45**	**37**
PVQ w/ dropout	1.26	**0.45**	36
PVQ	1.66	0.49	36

E. Mutual Information of State-of-the-art Codecs

We provide here additional insights into various state-of-the-art speech and music codecs. For all these codecs, we compute the mutual information between *individual* codebooks and all the remaining codebooks.

Music Codecs

We include in Figure 4 the mutual information of codes computed on the public music dataset FMA-Pop proposed in [4], as we found out that MusicCaps did not provide enough samples for reliable joint density histogram computation. Our results seem to show that both the original EnCodec (EnCodec-24kHz, (Défossez et al., 2023)) and the 4-level MusicGen variant of EnCodec (EnCodec-32kHz, (Copet et al., 2023)) suffer from relatively high inter-codebook

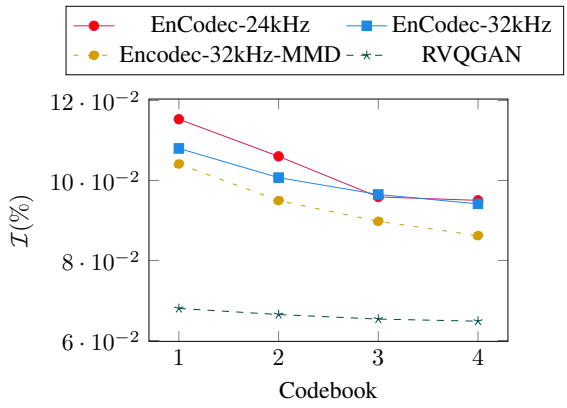

Figure 4. Mutual information between individual codebooks (on the horizontal axis) and all other codebooks, for difference codecs on FMA-Pop (Gui et al., 2024).

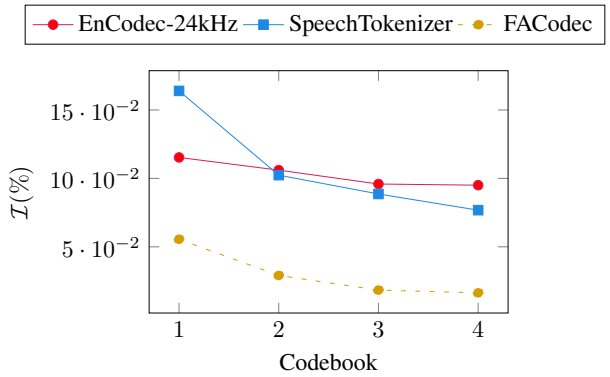

Figure 5. Mutual information between individual codebooks (on the horizontal axis) and all other codebooks, for difference codecs on LibriSpeech.

dependence, and that indeed RVQGAN obtains a large decrease of mutual information between codebooks, which can arguably be attributed to the choice of lower codebook dimensionality as suggested by the authors (Kumar et al., 2024). However, this does not mean that there is no room for improvement on this basis, as the independence- promoting mechanism for RVQGAN is structural, based on limitation of the amount of information learnable by a single codebook, and can also be completed with explicit MMD optimization, as we have demonstrated in Appendix B.

Speech Codecs

We compute the mutual information between the codebooks of SpeechTokenizer (Zhang et al., 2024) and FACodec (Ju et al., 2024) on LibriSpeech using 32k 200-second-samples and show the results on Figure 5. We compared to the results of the original EnCodec (EnCodec-24kHz, (Défossez et al., 2023)) which was trained on audio data including speech).

We observe that the mutual information between EnCodec and SpeechTokenizer codebooks and the other codebooks decrease monotonously with the codebook index, which is expected given the residual quantization scheme. For SpeechTokenizer we observe that the mutual information between the first codebook and the remaining codebooks is by far the largest across codebooks. Indeed, although the information in codebook 1 is specifically distilled from HuBert, there is actually no mechanism (unlike FACodec) that specifically prevents the codebooks 2:8 to use information from codebook 1. Yet, the authors confirm experimentally that the speaker-specific information is contained in the codebooks 2:8 and that codebook 1 contains mostly content information. This poses the question of how mutual information is exactly related to such semantics. For FACodec, the mutual information between the prosody stream and the content stream is also relatively high, but the mutual information between all other pairs of streams is very low, which shows some successful disentanglement. Overall it seems FACodec boasts the best level of disentanglement among the considered baselines. However, one must mention that speech semantic are much easier to investigate via the use of explicit audio properties (F0, phoneme label, ...) as opposed to music semantics. This enables for instance FaCodec to use gradient-reversal layers for supervising the disentanglement of their streams such as e.g. prosody and timbre. Our independence-promoting method, on the other hand, is fully unsupervised and domain-agnostic.

F. MUSHRA-style MOS Listening Test

Our subjective benchmark is a MUSHRA-style MOS listening test produced with the `webMUSHRA`[8] tool with `pymushra`[9] server management. In total, 12 annotators are asked to rate on a scale of 0 to 100 the overall quality of 12 10-second samples, whose descriptions were taken at random from the MusicCaps test set. All samples are normalized at -14dB LUFS(ITU-R, 2017). All annotators have a solid background either in audio or music processing. The instructions given on the training page are as follows: *"You are asked here to rate the different samples provided with respect to the reference. The rating should reflect the overall quality, comprising music quality, harmonic, melodic and rhythmic structure. You are not asked to rate the distance of the samples with respect to the reference in terms of sound similarity but along the aforementioned dimensions (quality, structure, consistency)."* The presentation order of the samples is randomized for each listener differently, and all 12 listeners listened to all of the samples. A snapshot of the interface for a randomized trial is shown on Figure 6. Inspired by the CrowdMOS guidelines, we excluded the annotations where reference track was rated below 85. We further excluded one annotator that systematically rated all

[8]https://github.com/audiolabs/webMUSHRA
[9]https://github.com/nils-werner/pymushra

generated samples below 50, resulting in the number of 11 annotators reported in the main paper.

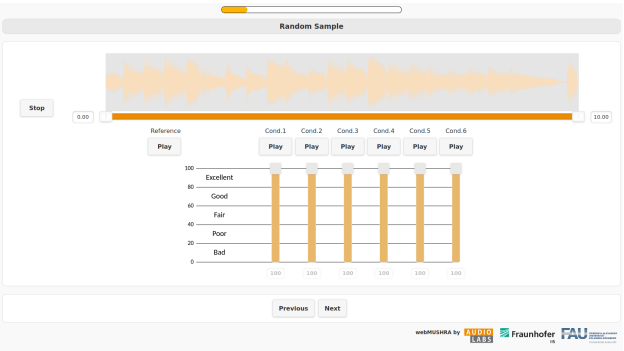

Figure 6. The MUSHRA listening test interface. Annotators listen to each sample and adjust the vertical bar on a continuous scale between 0 and 100. The reference track is given on the left and also hidden among the samples for rating.

G. MOS Evaluation with Amazon Mechanical Turk

We conducted a second subjective evaluation using the same subjective benchmark as (Copet et al., 2023; Kreuk et al., 2023), inspired by (Yang et al., 2023b). Human raters are sollicited via the Amazon Mechanical Turk platform and receive compensation meeting the American minimum wage. They assess two primary aspects of the audio signal: (i) overall quality (OVRL.), rated as the perceptual quality on a scale of 1 to 100; (ii) relevance to the text input (REL.), rated as the alignment between the audio and the text prompt on a scale of 1 to 100. Subjects evaluate 100 randomly selected files from the MusicCaps and AudioCaps test set, for music generation and general audio generation respectively. Each sample is assessed by at least 5 raters. The CrowdMOS[10] package is employed to filter out noisy annotations and outliers. This involves the exclusion of annotators who did not listen to the full recordings, those who rated the reference recordings below 85, and other CrowdMOS guidelines (Ribeiro et al., 2011). Results are shown in Table 6, and show that our method MusicGen-MMD is still ranking very high among baselines in terms of subjective ratings. However, the differences between the methods are rather marginal. The main difference between the methodology of the two tests resides in the recruitment of subjects (which is specified by the MUSHRA ITU-R BS.1534-0 recommendation). For the MUSHRA-style MOS experiment reported in the paper, we recruited confirmed audio listeners, and made sure that their setup was reliable (quiet environments, high-quality noise-canceling headphones...). On the other hand, we did not have any insight in the setups that subjects used in the MOS listening test in appendix. It is rather common than Mechanical Turk raters have low-quality setups, in potentially noisy environments, are not trained audio experts, and have little incitement for performance due to the low monetary retribution. For this reason, we believe the MUSHRA-style MOS evaluation reported in Table 1 is more reliable as the one conducted with Mechanical Turk raters, and therefore reported the first one in the main paper, and the second one in this appendix out of completeness.

Table 6. Subjective evaluation for text-to-music generation on MusicCaps. Mustango was trained on an augmented version of MusicCaps, therefore we put it aside the other baselines. Mean and 95 % confidence intervals are showed. The samples presented were 10 second long sampled at 16kHz, which matches the training conditions of Mustango, AudioLDM and AudioLDM2-Music. In comparison MusicGen and MusicGen-MMD were trained on 30 second-long segments sampled at 32kHz.

Model	# params	OVRL. ↑	REL. ↑
Ground-Truth	-	*92.49 ± 1.65*	*92.89 ± 1.38*
Mustango	1.4 B	*81.24 ± 2.43*	*84.27 ± 1.95*
AudioLDM	416 M	**84.70 ± 2.25**	84.20 ± 3.12
AudioLDM2-Music	347 M	81.93 ± 2.01	84.91 ± 2.55
MusicGen	300 M	84.52 ± 2.19	85.11 ± 1.98
MusicGen-MMD (ours)	300 M	84.18 ± 1.74	**87.57 ± 2.16**

[10] http://www.crowdmos.org/download/