Applying textual inversion to control and personalize text-to-music models

Carl Thomé[1 [0000−0002−8225−5191]], Bob L. T. Sturm[1 [0000−0003−2549−6367]], John Pertoft[1 [0009−0008−7717−2261]], and Nicolas Jonason[1 [0009−0003−8553−3542]]

KTH Royal Institute of Technology, Stockholm, Sweden

Abstract. A text-to-music (TTM) model should synthesize audio that reflects the concepts in a given prompt as long as it has been trained on those concepts. If a prompt references concepts that the TTM model has not been trained on then the audio it synthesizes will likely not match. This paper investigates the application of a simple gradient-based approach called *textual inversion* (TI) to expand the concept vocabulary of a trained TTM model without compromising the fidelity of concepts on which it has already been trained. We apply this technique to MusicGen and measure its reconstruction and editability quality, as well as its subjective quality. We see TI can expand the concept vocabulary of a pretrained TTM model, thus making it personalized and more controllable without having to finetune the entire model.

Keywords: Text-to-music · Textual inversion · audio reference.

1 Introduction

Text-to-music (TTM) models have received significant attention with the emergence of large-scale pretrained models such as MusicGen [3], MusicLM [1], AudioLDM [10], Moûsai [19], Stable Audio [5], and JEN-1 [9]. These models are able to synthesize plausible music audio from text prompts, offering a wide range of applications from music composition and music production to live streaming music synthesis of text sources as a novelty. However, the ability to control and personalize a TTM model remains a challenging task, particularly when attempting to express ideas that do not clearly relate to its concept vocabulary. If the model has not learned a particular concept in a prompt then it will not produce a relevant output. How might the concept vocabulary of an existing TTM model be increased, thus personalizing it and increasing its controllability?

There are multiple ways one may be able to increase the controllability of TTM generation, ranging from finetuning the whole model [16], to adding adapter layers with feature matching losses [15, 20], to zero-shot editing techniques [14], and an approach we explore here called Textual inversion (TI). TI is a simple gradient-based method that has been applied to expand the concept vocabulary of pretrained text-to-image models [6]. By providing a small set of images, a model can be taught new "words" without affecting the concepts on which it has already been trained. It should be emphasized that TI does not change the original

parameters of a model, but rather adds new parameters to its embedding layer and trains only those. This is attractive as it circumvents risks of catastrophic forgetting [13], and means that resource-constrained environments do not need to juggle multiple sets of model weights — which is a practical challenge in terms of memory and disk space usage within a multi-tenant application deployment.

In this work, we explore TI as a means to control and personalize by audio reference a specific TTM model: MusicGen [3], a language-model approach to music generation. DreamSound [16] also considers the personalization of TTM models using TI, but by latent diffusion models — specifically AudioLDM [10]. They conclude that TI does not sound as good as tuning the entire model [18]. This is corroborated by emerging best practices in online communities focused on using text-to-image models [2], where a commonly held belief is that tuning the whole model attains higher quality. However, tuning the entire model comes with a high computational cost, and requires one model per concept in the vocabulary, which can be avoided with TI. We thus look at how well TI can perform for controlling and personalizing a language-model approach to TTM.

2 Personalizing MusicGen with TI

Figure 1 shows the procedure of adapting MusicGen to a new concept using TI. MusicGen [3] is a language-model approach to TTM generation that consists of three parts: an audio model A, a text encoder T and a music model M. First, A converts audio into a token sequence by an autoencoder called EnCodec [4]. Second, T produces another token sequence from textual descriptions of the music with an encoder-decoder model called T5 [17]. The audio tokens are autoregressively modelled by M, conditioned on the text tokens. The text token S^* represents a concept which we want to introduce to MusicGen. We create synthetic prompts using the concept with neutral text prompts, such as "A recording of a S^*", or "The sound of S^*", which are linked with an audio reference x of the target concept. TI involves gradient descent optimization of new parameters added to the embedding matrix of T using these text-audio pairs.

Denote the new parameters as v. Given y as neutral text involving S^*, the original TI optimization [6] was defined for latent diffusion models as

$$v^* = \underset{v}{\operatorname{argmin}} \mathbb{E}_{z\sim\mathcal{E}(x)y,\epsilon\sim\mathcal{N}(0,1),t}\Big[\|\epsilon - \epsilon_\theta(z_t, t, T(y; v))\|_2^2\Big] \tag{1}$$

where the diffusion model ϵ_θ is fixed.[1] In the case of a language-model approach like MusicGen [3] the optimization instead becomes minimizing the categorical cross-entropy between left-to-right shifted audio tokens as

$$v^* = \underset{v}{\operatorname{argmin}} \sum_{k=1}^{K} C_k(\sigma(A(x)), M(A(x), T(y; v))) \tag{2}$$

[1] See Luo et. al [12] for full details on latent diffusion models.

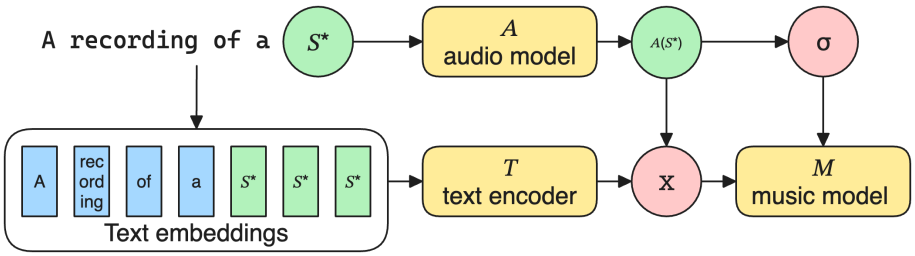

Fig. 1. Overview of how we apply TI. The audio concept $S*$ is optimized to minimize the music model's next-token prediction. Text and audio embeddings are combined by cross-attention X. Target audio tokens are shifted left-to-right by σ and masked to formulate an autoregressive training setup. All parameters are kept fixed during finetuning except for the concept embeddings.

where C_k is cross-entropy loss for the k:th codebook in the RVQ-VAE. It is important to note that σ shifts the audio tokens $A(x)$ by one position to the right, and applies causal masks by a delay pattern for codebook interleaving [3]. This is what constitutes the autoregressive nature of the model.

There are pretrained MusicGen models available with different sizes and input conditioning. We use the 3.3-billion parameter model without melody conditioning. The motivation for this choice is that the melody-conditioned models prepend text tokens to the audio token sequence, rather than performing cross-attention at each step. In preliminary experiments we found that prepending leads to ineffective control. We create ten concept tokens in the T5 tokenizer by appending new ids after its 32100 tokens. Each token corresponds to one row in the text embedding weight matrix of T with values initialized by sampling from a multivariate Gaussian distribution. The parameters of this distribution are set to the empirical mean and variance of the pretrained T5-base model's text embeddings (the same trained T5 that was used in MusicGen pretraining). With a T5 encoder embedding size of 768, and ten embeddings to represent the concept, the number of trainable parameters is thus 7680 weights, which embed 240 ms of digital audio at 32 kHz.

The concept parameters are then finetuned with equation 2 as the loss function, with two-second audio clips forming minibatches of 100 audio tokens per example. The text was synthesised with a set of 31 neutral text prompts, with a padding token and length truncation at 512 tokens. With a RTX 3090 the finetuning takes approximately 10 minutes per concept using 1000 steps and a batch size of 32. We use the AdamW optimizer [11] with a learning rate warm-up and exponential decay annealing schedule of

$$l_t = \begin{cases} l_{t-1} \times \frac{1}{10^{10-t}} & \text{if } t < 10 \\ l_{t-1} \times \gamma & \text{else} \end{cases} \tag{3}$$

with $\gamma = 0.99$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and L1 weight decay of 0.01. After training we use an exponential moving average of the concept embeddings with a weight

decay of 0.95 [8]. It is important to note that only the embeddings of the new concept are optimized, and the rest of the model parameters are kept fixed. However, the loss function is still computed by going through MusicGen one forward pass and backpropagating gradients to the new T5 encoder embeddings.

At test-time we sample half of the concept tokens to balance the influence strength of the new concept in a prompt. If every concept token is used, the surrounding tokens are ignored in the output. Thus to make sampling be invariant to the T5 positional embedding we randomly shuffle them in each synthetic text during training. We also regularize the cosine similarity of the new concept embeddings with repeatedly sampled T5 embeddings, with the rationale being that a new concept should be orthogonal to the concepts already trained into the model, and should not share characteristics with most text tokens, e.g. "Newspaper" or "Sandwich".

3 Evaluation

We use the demo audio/text examples from DreamSound [16] for our evaluation. First, we compute the average pairwise audio distance between a finetuned concept's audio output and its reference audio. We also compute the distance between text prompts and the resulting audio. These distances are specifically contrastive language audio pretraining (CLAP) scores that we produce by running data through a jointly optimized text and audio encoder to produce text embeddings E^t and audio embeddings E^a. These encoders were pretrained by Wu et. al [21] with a contrastive learning paradigm over audio/text pairs from AudioSet [7] and LAION-Audio-630k [21]. By computing the mean distance $d(E^a, E^t)$ between the resulting embeddings, we get a data-driven notion of agreement between text and audio, or audio and audio depending on input type. Second, we conduct a listening study asking subjects to select which of two audio syntheses (total of ten pairs) best matches a prompt. Each subject is also offered to skip when they cannot decide, and "Neither" when they perceive both outputs as too dissimilar to the description. We use two different kinds of prompts: "A recording of a S*" is called "reconstruction", and "A disco song with a S*" is called "editability". We tabulate the number of times each system output is selected and compute the proportions. All audio stimuli are mono mixed, volume normalized to -24 dB LUFS, trimmed to ten seconds' duration and sampled at 32 kHz.

Figure 2 shows that the average test scores are comparable. Reconstruction appears to be higher for AudioLDM and DreamSound than textual inversion with MusicGen. The editability score is somewhat higher for MusicGen than for AudioLDM, indicating that text and audio matches somewhat better. In terms of central tendency, spread, and distribution we mostly observe similarities, while we hear clear differences in audio characteristics between systems in listening tests, indicating limitations of CLAP scores.

Table 1 shows the results of the listening study, which involved 26 people selected by convenience sampling. The participants had varying degrees of music experience, including professional music producers, music information retrieval

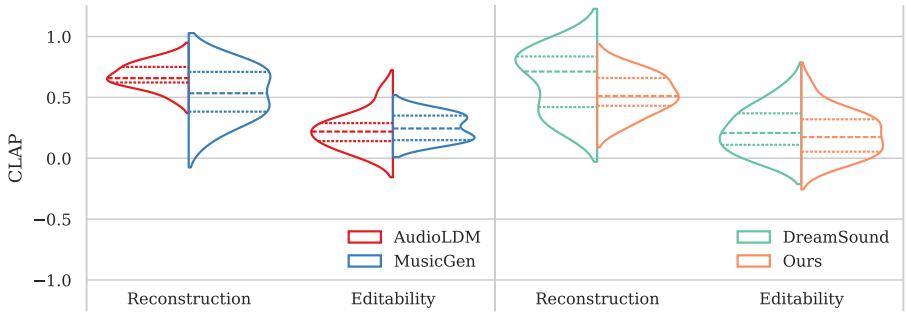

Fig. 2. (left) Comparison of CLAP scores for TI applied to AudioLDM and MusicGen. Reconstruction is the average pairwise cosine similarity between CLAP audio embeddings of reference and reconstruction audio excerpts. Editability is the average pairwise cosine similarity between CLAP text embeddings of editability text prompts and the system audio output. (right) Comparison of CLAP scores between DreamSound [16] and our system. The interior lines show quartiles of the data per group, with dashed being the median and dotted being 25% and 75% percentiles.

researchers, and casual music listeners. We see that participants generally found the audio output of TI applied MusicGen to match the description best when it comes to editability, but DreamSound for reconstruction. Note that this listening study is merely a preliminary counterpart to gauge the sanity of the results of the comparing CLAP scores. As for any application of computational creativity, the best method of evaluation is by judging for oneself through interaction with the system. We thus invite the reader to listen to output examples of these systems.

Table 1. Listening study results comparing TI with DreamSound and MusicGen.

	Editability	Reconstruction
DreamSound	41.3%	49.4%
MusicGen	44.2%	37.8%
Neither	14.4%	10.9%
Undecided	0.0%	1.9%

4 Discussion

Participants in the listening study remarked that MusicGen appears to provide more varied output sequences that do not follow the description's audio dynamics and melody precisely. This is possibly due to how its prediction is formed autoregressively, while AudioLDM iteratively refines the entire ten second sequence by its reverse diffusion process. Thus the diffusion approach is more strictly following

the reference audio structure. Whether this is a desirable property of a TTM control mechanism depends on the intended application but we believe both modes could become complementary for music creation. Since we have heard musical examples during experiments that blend a finetuned concept in a genre prompt convincingly, it would be interesting to understand particularly *when* TI with a TTM model based on a language-model approach works well or not. Extending this investigation to include additional test concepts of different types of instruments and styles, relying on more test subjects for improved statistical analysis, and investigating other TTM models such as MusicLM [1] would be interesting.

It is noteworthy that pretrained models for MusicGen were made publicly available as creating such a model is prohibitively expensive, and exploration would otherwise be inaccessible to many researchers. Other TTM models remain unavailable to the public and there is no standard way of running them so considerable work is involved in order to include multiple models in studies. While we cannot freely access private models, we encourage the research community to collaborate on a shared reconstruction/editability dataset for subject-driven music generation, and to start comparing approaches for controlling and personalizing TTM models in a standardized manner for easy comparison. With that said, we know that automatically judging perceived quality of machine generated music is challenging. As our goal is to correlate well with human enjoyment of music creation, we need to acknowledge that the metrics and concepts used in this work are not necessarily the ones that should be optimized fully on leaderboards. It remains important to be mindful of eventual limitations of what information they convey. The underlying encoders for CLAP scores were trained on specific music and language data from specific data sources [21], and the test example concepts were taken from DreamSound's [16] demo page rather than an open benchmark dataset. Despite this, the subjective assessment seem promising and we recommend readers to listen to audio examples.

Acknowledgement. This work was supported by the European Research Council under the European Union's Horizon 2020 research and innovation programme (MUSAiC project, Grant agreement No. 864189).

References

1. 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., Frank, C.: MusicLM: Generating music from text (Jan 2023)
2. AUTOMATIC: Stable diffusion web UI
3. Copet, J., Kreuk, F., Gat, I., Remez, T., Kant, D., Synnaeve, G., Adi, Y., Défossez, A.: Simple and controllable music generation (Jun 2023)
4. Défossez, A., Copet, J., Synnaeve, G., Adi, Y.: High fidelity neural audio compression (Oct 2022)
5. Evans, Z., Carr, C.J., Taylor, J., Hawley, S.H., Pons, J.: Fast Timing-Conditioned latent audio diffusion (Feb 2024)

6. Gal, R., Alaluf, Y., Atzmon, Y., Patashnik, O., Bermano, A.H., Chechik, G., Cohen-Or, D.: An image is worth one word: Personalizing Text-to-Image generation using textual inversion (Aug 2022)
7. Gemmeke, J.F., Ellis, D.P.W., Freedman, D., Jansen, A., Lawrence, W., Channing Moore, R., Plakal, M., Ritter, M.: Audio set: An ontology and human-labeled dataset for audio events (2017)
8. Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D., Wilson, A.G.: Averaging weights leads to wider optima and better generalization (Mar 2018)
9. Li, P., Chen, B., Yao, Y., Wang, Y., Wang, A., Wang, A.: JEN-1: Text-Guided universal music generation with omnidirectional diffusion models (Aug 2023)
10. Liu, H., Chen, Z., Yuan, Y., Mei, X., Liu, X., Mandic, D., Wang, W., Plumbley, M.D.: AudioLDM: Text-to-Audio generation with latent diffusion models (Jan 2023)
11. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization (Nov 2017)
12. Luo, C.: Understanding diffusion models: A unified perspective (Aug 2022)
13. Luo, Y., Yang, Z., Meng, F., Li, Y., Zhou, J., Zhang, Y.: An empirical study of catastrophic forgetting in large language models during continual fine-tuning (Aug 2023)
14. Manor, H., Michaeli, T.: Zero-Shot unsupervised and Text-Based audio editing using DDPM inversion (Feb 2024)
15. Novack, Z., McAuley, J., Berg-Kirkpatrick, T., Bryan, N.J.: DITTO: Diffusion Inference-Time T-Optimization for music generation (Jan 2024)
16. Plitsis, M., Kouzelis, T., Paraskevopoulos, G., Katsouros, V., Panagakis, Y.: Investigating personalization methods in text to music generation (Sep 2023)
17. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer (Oct 2019)
18. Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., Aberman, K.: DreamBooth: Fine tuning Text-to-Image diffusion models for Subject-Driven generation (Aug 2022)
19. Schneider, F., Kamal, O., Jin, Z., Schölkopf, B.: Moûsai: Text-to-Music generation with Long-Context latent diffusion (Jan 2023)
20. Wu, S.L., Donahue, C., Watanabe, S., Bryan, N.J.: Music ControlNet: Multiple time-varying controls for music generation (Nov 2023)
21. Wu, Y., Chen, K., Zhang, T., Hui, Y., Berg-Kirkpatrick, T., Dubnov, S.: Large-scale contrastive Language-Audio pretraining with feature fusion and Keyword-to-Caption augmentation (Nov 2022)