TINYMUSICIAN: ON-DEVICE MUSIC GENERATION WITH KNOWLEDGE DISTILLATION AND MIXED PRECISION QUANTIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

The success of the generative model has gained unprecedented attention in the music generation area. Transformer-based architectures have set new benchmarks for model performance. However, their practical adoption is hindered by some critical challenges: the demand for massive computational resources and inference time, due to their large number of parameters. These obstacles make them infeasible to deploy on edge devices, such as smartphones and wearables, with limited computational resources. In this work, we present *TinyMusician*, a lightweight music generation model distilled from MusicGen (a State-of-the-art music generation model). TinyMusician integrates two innovations: (i) Stage-mixed Bidirectional and Skewed KL-Divergence and (ii) Adaptive Mixed-Precision Quantization. The experimental results demonstrate that TinyMusician retains 93% of the MusicGen-Small performance with 55% less model size. TinyMusician is the first mobile-deployable music generation model that eliminates cloud dependency while maintaining high audio fidelity and efficient resource usage. [1]

1 INTRODUCTION

Music, reflecting culture, social classes, ethnic identities, and historical eras, has woven itself into humanity's shared heritage through centuries of evolution (Toynbee, 2012). Today, artificial intelligence (AI) has demonstrated remarkable breakthroughs across multimodal domains, including image generation (Liu, 2023), inpainting, outpainting (Silva & Oliveira, 2024), short video production (Sun, 2024), etc.

Large-Language Models (LLMs) (Liu et al., 2023; Bi et al., 2024) showed excellent modeling capabilities in obtaining complex relationships in long-term contexts, which the music genre inherited. In view of this, MusicLMs (Agostinelli et al., 2023) and many subsequent works (Copet et al., 2023; Lam et al., 2023; Suno-Ai, 2023) successfully applied LLMs in music generation, capitalizing on their ability to capture intricate patterns in musical sequences.

However, the pursuit of higher-quality AI music generation, driven by two technical imperatives, Scaling Law (Kaplan et al., 2020) and Emergent Capability (Berti et al., 2025), has led to a surge in model parameters, creating critical challenges in both computation and deployment. The frequently discussed text-to-music models, for example, MusicGen-Large (Copet et al., 2023) and YuE-7B (Yuan et al., 2025), having undergone training on large-scale datasets, exhibited excellent capabilities in synthesizing music (Austin et al., 2021). The generated music was characterized by high fidelity (Yao et al., 2025), a high level of accuracy and detail in its sound quality, and a strong coherence with the provided text prompts. Yet, this parameter escalation introduces a trilemma between musical fidelity, computational cost, and deployment feasibility, especially for on-device deployment, where there are limited computational resources, such as smartphones and extended reality glasses. The dependency on a cloud, with high computational overheads, hinders the proliferation of AI-generated products into real-world applications, such as games, and keeps these models server-dependent (Rawassizadeh et al., 2018; Huang et al., 2024; Rawassizadeh & Rong, 2023).

[1] [URL for code and model download is anonymized for peer review.]

There are several efforts to reduce the model size, especially in Transformer architecture, such as efficient self-attention mechanisms (Tian et al., 2025). Furthermore, several approaches such as Mixture of Experts (MoE) (Jacobs et al., 1991), and Low-Ranked Adaptation (LoRA) Hu et al. (2021) are used to reduce the computational cost of feed-forward layers of transformer architecture.

On the other hand, several approaches focus on enabling the neural network models' deployment on consumer electronics, such as Federated Learning (Li et al., 2020) and model compression techniques. Three common approaches focus on model compression and reducing the size of a neural network, including Knowledge Distillation (Gou et al., 2021), pruning (Zhang et al., 2022), and quantization (Wei et al., 2024).

In short, knowledge distillation transfers knowledge from a teacher model (baseline model) to a student model (smaller model), which ensures fidelity of results while having a smaller number of model parameters. This technique has demonstrated its efficacy across multiple AI domains. For instance, Mullapudi et al. (2019) proposed JITNet, employing MRCNN (Tian et al., 2019) as the teacher model, and reduced the number of parameters from 300 million to 7 million. Sun et al. (2019a) introduced Patient Knowledge Distillation, which distills the 12-layer BERT-original (Devlin et al., 2019) into a 6-layer BERT while preserving 97% of its performance.

In addition to the knowledge distillation, neural network quantization has emerged as another paradigm of critical model compression (Gou et al., 2021; Wei et al., 2024). By converting high-precision model weights into compact low-bit representations without altering the network architecture (Gholami et al., 2022), this technique has been proven effective in domains such as audio processing (Derrien et al., 2006), Deep Reinforcement Learning (Lu et al., 2024) and image processing (Rokh et al., 2023).

The third common approach is pruning (Zhu & Gupta, 2017), which includes removing neurons that are not contributing much to the final output. Pruning could be determined based on the weight, activation value, or even the entire neuron and its connections (Rawassizadeh, 2025).

While these techniques have been extensively explored in different fields, especially image recognition (Rokh et al., 2023) and natural language processing (Sun et al., 2019a), their application to music generation remains underexplored.

In this work, we present *TinyMusician*, a novel lightweight model for mobile, on-device music generation, distilled from the state-of-the-art MusicGen-Small (Copet et al., 2023) architecture, integrating Stage-mixed Bidirectional KL-divergence in Knowledge Distillation with temperature annealing strategy (Zhang et al., 2024) to enhance knowledge transfer fidelity between teacher and student models. To further enhance inference efficiency, we also implement adaptive mixed-precision quantization (Chauhan et al., 2023) and achieve 55% reduction in model size compared to the original MusicGen with 9.5% sacrificing melodic or harmonic fidelity.

Figure 1 presents the architecture of TinyMusician. We have integrated TinyMusician into the iOS mobile platform through ONNX runtime conversion and platform-specific optimization for iOS and Android. To our knowledge, this is the first on-device music generation model that can run on smartphones independently of the cloud or other large computational resources.

2 Related Works

Deploying high-fidelity music generation on edge devices faces two co-dependent barriers: (1) Resource-Intensive Models: Transformer-based music models (Table 1) achieve remarkable quality but require strong GPU resources, which are incompatible with edge devices' constraints; (2) Compression Limitations: Existing compression techniques lack specialized mechanisms to preserve musical fidelity, risking perceptual degradation. Therefore, our related work is composed of two sections: synthetic music generation and model compression.

2.1 Synthetic Music Generation

Previous to neural network advances, the Markov chain (Hassani & Wuryandari, 2016; Shapiro & Huber, 2021), rule-based models (Hastuti et al., 2017; Sneyers & De Schreye, 2010), and evolutionary algorithms (Loughran & O'Neill, 2020; Kaliakatsos-Papakostas et al., 2020) are three main

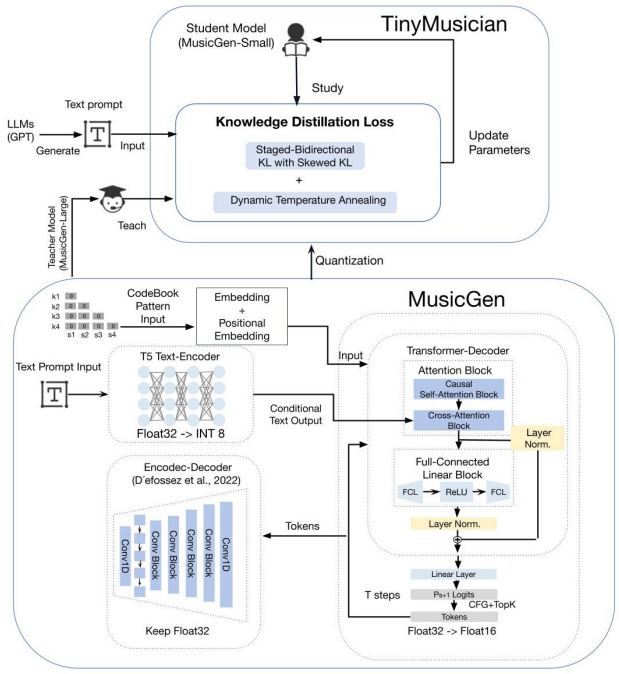


Figure 1: The architecture of TinyMusician with respect to its teacher model, i.e., MusicGen small.

groups of methods that are mainly used to generate music. These methods are typically parameter-based, requiring human input of parameters or configurations to guide music generation. Music generated by these methods remained quite limited.

Later, with the development of deep neural networks, generative models show incredible ability in sequential data construction, including music. Several types of generative models have been developed to meet the high-quality requirement of music generation, including RNN models (Goel et al., 2014; Dua et al., 2020), GAN based-models (Zhang et al., 2021; Huang & Huang, 2020), and VAE based-models (Dhariwal et al., 2020; Liang et al., 2019). For example, Jukebox (Dhariwal et al., 2020), one of the first VAE-based models, can generate full vocal music. Although the quality is limited and slow, it still demonstrates the ability to produce music that aligns with the inputs of the lyric, artist, and Genre.

Recently, diffusion models (Song et al., 2020) and transformer-based models (Kang et al., 2024; Shih et al., 2022) have emerged as the mainstream in music generation models. The learning process of diffusion models involves two core steps: a forward process that gradually adds noise to a sample and a reverse process that aims to denoise and reconstruct the original data (Rawassizadeh, 2025). ERNIE-Music (Zhu et al., 2023) is a diffusion-based architecture specifically designed for music, and it involves a forward step of gradually adding Gaussian noise to music waveforms and a reverse denoising process to reconstruct the original audio, using a U-Net with conditional self-attention (Ibtehaz & Rahman, 2020) to integrate text prompts from an ERNIE-M text encoder for direct text-to-waveform translation.

Transformer-based models (Wen et al., 2022), on the contrary, are experts in modeling long-range dependencies and structural patterns, such as melodic repetition, harmonic progression, or rhythmic patterns, by processing musical sequences as tokenized events (notes, pitches, instruments) with positional encoding (Dash & Agres, 2024). However, the strong performance of transformer models also comes with a tradeoff: the transformer-based architecture requires substantial computational resources, which hinders their deployment on small battery-powered devices.

Since our approach is also transformer-based, the popular transformer models for music generation are listed in Table 1. As shown in this Table, even small state-of-the-art models have a large number of parameters. For example, Yue-7B (Yuan et al., 2025) has 7 billion parameters and demands

Model	Model Size	Params	GPU (Inference)
MusicGen-Small (Copet et al., 2023)	2.36GB	300M	RTX 2060 6GB
Music-LM (Agostinelli et al., 2023)	3.44GB	860M	RTX 3050 8GB
YuE-7B (Yuan et al., 2025)	13GB	7B	RTX 3090 24GB
Flux (Fei et al., 2024)	8.44GB	2.1B	RTX 3090 24GB
Musictango (Melechovsky et al., 2023)	5.6GB	1.4B	RTX 3060 12GB
Spectrogram (Hawthorne et al., 2022)	1.65GB	412M	RTX 2060 6GB

Table 1: Music generation model sizes along with GPU memory utilization

about 40GB of memory model storage. Even the smallest model, MusicGen-Small, demands 10GB of GPU memory and an RTX 3080 GPU to achieve acceptable inference speeds. Large model parameters and high computational costs require devices with very high memory and computing power.

These evidences show that directly deploying such models not only incurs significant resource costs but also triggers long inference times. Additionally, deploying these models on edge devices such as mobile phones is impossible due to their limited storage and computing resources.

2.2 Model Compression

A reasonable model compression method finds the balance between compressed pre-trained model memory and model performance so that the model can be deployed on various resource-constrained devices (Tang et al., 2024).

Quantization methods have some advantages over Pruning and Knowledge Distillation. In particular, first, they are cost-effective, and most of the quantization methods don't need to retrain the entire model, making them easier for researchers with limited computing resources. Second, they support effective compression, because the weights of models from 32-bit Float to 8-bit or 4-bit Int could drastically compress model size to approximately 1/4 or 1/8. Third, quantization is highly compatible with most other model compression methods and thus flexible.

Quantization-aware training (QAT) and Post- Training quantization (PTQ) is a common Quantization method (Rawassizadeh, 2025). QAT (Esser et al., 2019) aims to quantize the model during the training phase, while PTQ (Shang et al., 2023) considers the quantization after training. Due to the cost-effectiveness of time and computational resources, PTQ is more popular. Most PTQ approaches quantize parameters in weights and activations in each layer, and can be divided into three subsets: Weight-only Quantization, Key-Value (KV) Cache Quantization, and Weight-activation Quantization (Liu et al., 2025). PTQ advancements have reshaped model efficiency. For example, GPTQ (Frantar et al., 2022) is a Weight-only method that can compress popular open-source models down to 3 and 4 bits. SmoothQuant (Xiao et al., 2023), in contrast, introduces joint weight-activation quantization, balancing their dynamic ranges to reduce error propagation in vision transformers. While PTQ methods have improved model efficiency, their application to music generation models remains underexplored.

Unlike text or images, music synthesis demands precise preservation of temporal dynamics and spectral fidelity, which are highly sensitive to quantization errors in both weights and activations. Applying uniform quantization across all model weights and activations, as commonly done in other domains, risks significant degradation in musical quality (Lohar et al., 2023). To address these challenges, for music generation, a mixed-precision quantization approach is essential. In contrast, Xiao et al. (2023) introduce joint weight-activation quantization, balancing their dynamic ranges to reduce error propagation in vision transformers. While PTQ methods have improved model efficiency, their application to music generation models remains underexplored. Unlike text or images, music synthesis demands precise preservation of temporal dynamics and spectral fidelity, which are highly sensitive to quantization errors in both weights and activations.

Applying uniform quantization across all model weights and activations, as commonly done in other domains, risks significant degradation in musical quality. To address these challenges, a mixed-precision quantization approach is essential, and our model, *TinyMusician* implements it.

Knowledge Distillation (KD), based on how to design the loss function, can be further categorized into two groups: logit-based KD, and feature-based KD (Hinton et al., 2015). Logit-based KD typically uses KL-Divergence or Mean Square Error (MSE) to minimize the logits between the teacher and the student. DistilBERT (Sanh et al., 2019), for example, is a KD of BERT, which is 40% smaller, 60% faster, retains 97% of BERT's language understanding capabilities, and is trained with a triple loss during pre-training, demonstrating its effectiveness in various downstream tasks. Schmid et al. (2023) propose an offline KD training method from high-performance yet complex Transformer models to efficient CNN models. It constructs different audio tagging models with different complexities, outperforming previous solutions in terms of model size, computational efficiency, and prediction performance, assessed via Frechet Audio Distance (FAD) (Kilgour et al., 2018), which quantifies audio by comparing feature distributions, and CLAP scores (Ye et al., 2023), which measure text-audio semantic alignment via contrastive learning, and achieving a new single-model state-of-the-art mean average precision of 0.483 on the AudioSet dataset.

Feature-based KD aims to minimize the intermediate features between the teacher and the student. PKD (Sun et al., 2019b) introduced MSE as a loss function and proposed two strategies: the student learns the last few layers in the teacher, and the others learn every two layers' representations of the teacher. MT-BERT (Wu et al., 2021) method uses multiple teacher pre-trained language models with a new finetuning framework and new loss functions to better compress PLMs and outperforms single-teacher and some multi-teacher distillation methods. However, as with Quantization, KD is also still underexplored in the music generation area.

3 TINYMUSICIAN

As it has been stated before, in addition to knowledge distillation, TinyMusician introduces two salient novelties to enable on-device music deployment, which we describe in this section.

3.1 KNOWLEDGE DISTILLATION WITH STAGE-MIXED BI-DIRECTIONAL AND SKEWED KL

To perform knowledge distillation, we choose MusicGen-Large as the teacher model (Copet et al., 2023), and apply our knowledge distillation on MusicGen-Small, as the student model, and further improve it, which leads to the TinyMusician. Traditional one-directional KL-Divergence aims to force the student model to mimic the output distribution of the teacher model. However, music should keep chronological coherence and local tone detail; thus, inspired by the methodology proposed by Yang et al. (2025), we introduced an improved formulation of Bidirectional KL-Divergence, called ***Stage-mixed Bidirectional KL-Divergence***, as a loss function and conducted comparative experiments against traditional KL Divergence variants. The detailed analysis of different divergence metrics and experimental configurations will be presented in Section 4. The definition of Stage-mixed Bidirectional and Skewed KL-Divergence is presented in Equation 1.

$$\begin{aligned}\mathcal{L}_{\text{KL}}(t) = \alpha(t) \cdot [\gamma_1 D_{\text{KL}}(T \| S) + (1-\gamma_1) D_{\text{KL}}(T \| S_\lambda)] \\ + (1-\alpha(t)) \cdot [\gamma_2 D_{\text{KL}}(S \| T) + (1-\gamma_2) D_{\text{KL}}(S \| T_\lambda)]\end{aligned} \tag{1}$$

where the mixed distributions are defined as:

$$S_\lambda = \lambda T + (1-\lambda) S \tag{2}$$

$$T_\lambda = (1-\lambda) T + \lambda S \tag{3}$$

In Equation 1, $\mathcal{L}_{\text{KL}}(t)$ represents the stage-mixed KL-Divergence loss at time step t. $\alpha(t)$ is a dynamic weight function that varies with the time step t, which is used to adjust the proportion of different KL-Divergence terms in different stages. γ_1 and γ_2 are hyperparameters. γ_1 is used to balance the two KL-Divergence terms in the first part of the equation, and γ_2 is used for the second part. T represents the teacher distribution, and S represents the source distribution. $\mathcal{D}_{\text{KL}}(A||B)$ represents the KL-Divergence between distribution A and distribution B. The mixed distributions $S_\lambda = \lambda T + (1-\lambda)S$ and $T_\lambda = (1-\lambda)T + \lambda S$ represent convex combinations of the teacher and student distributions, where S_λ smooths the student's output for stable forward KL Divergence optimization, while T_λ robustifies the teacher's reference for resilient reverse KL Divergence learning.

$$\alpha(t) = \begin{cases} 1 & \text{if } t < \tau_{\text{step}} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

In Equation 4, $\alpha(t)$ is a dynamic weight function that depends on the current time step t and a predefined step threshold τ_{step}. When the time step t is less than the threshold τ_{step}, $\alpha(t)$ takes the value of 1, the first part of the loss equation, i.e., $[\gamma_1 \mathcal{D}_{\text{KL}}(T||S) + (1-\gamma_1)\mathcal{D}_{\text{KL}}(T||S_\lambda)]$ is taken into the account, while the second part, i.e., $[\gamma_2 \mathcal{D}_{\text{KL}}(S||T) + (1-\gamma_2)\mathcal{D}_{\text{KL}}(S_\lambda||T)]$ weights 0 and is thus ignored. When $t \geq \tau_{\text{step}}$, $\alpha(t)$ is 0. In this case, the first part of the loss equation is ignored, and the second part is fully taken into account.

To meet the requirement of KD in different stages, we design an *adaptive temperature annealing mechanism* inspired by the strategy proposed by Manvi et al. (2024). Unlike the exponential annealing schedule proposed in their work, our approach employs a linear decay. This adaptive temperature annealing is straightforward and scalable because it avoids the complex parameter tuning required by nonlinear schedules, while still effectively balancing exploration and exploitation in the generation process. Equation 5 formalizes our adaptive temperature annealing approach.

$$\tau = T_b - (T_b - T_f) \times \left(\frac{s}{L_{max}} \right) \tag{5}$$

Here, T_b represents the initial temperature, T_f represents the final temperature, s represents the current step, and L_{max} represents the maximum output length.

3.2 Customized Quantization

In addition to our proposed KD, we adopt a post-training (Shang et al., 2023) mixed-precision method to quantize the MusicGen-small model. The MusicGen model can be partitioned into three distinct components: the T5 Text-Encoder (Ni et al., 2021), the MusicGen-Decoder, and the Encodec-Decoder (Défossez et al., 2022). Each of these components is quantized into different formats: specifically, the Text-Encoder is quantized to Int8 to balance efficiency and representation preservation; the MusicGen-Decoder is quantized to Float16 to maintain autoregressive generation stability; and the Encodec-Decoder is kept in Float32 to ensure high-fidelity audio reconstruction.

The Text Encoder takes text as input embeddings, produced by a tokenizer or embedding layer, and outputs the last hidden states. In MusicGen-Decoder, the transformer performs autoregressive token generation by processing a sequence of discrete tokens, step-by-step. At each step, it uses causal self-attention to focus only on previously generated tokens, ensuring no future information is accessed. It also incorporates conditional signals (the text embeddings) via cross-attention to guide the generation. Based on these inputs, the transformer predicts the next token (using Classifier Free Guidance (CFG) (Sanchez et al., 2023) strategy and the top-k sampling strategy to guide the model's output) in the sequence, which is then added to the existing sequence. This iterative process continues until a complete token sequence is generated, and each new token builds on the context of all prior ones.

Lastly, the generated tokens are then fed into a subsequent Encodec-Decoder module. The decoder within this module further decodes these intermediate tokens into raw audio waveforms, completing the end-to-end text-to-music generation pipeline. Quantization efficacy and latency/quality tradeoffs are evaluated in Section 5, demonstrating minimal degradation compared to full-precision baselines.

4 Experiments

4.1 Dataset

We conduct our experiments on the MusicCap Dataset (Lee et al., 2023), which is a large-scale dataset for music-text alignment tasks. It consists of 5,500 high-quality music-text pairs. Each sample is annotated with two types of descriptions: (i) a list of English-language musical aspects, which captures elements such as genre, tempo, and instrumentation; and (ii) a free-form text caption authored by professional musicians, offering qualitative insights into the musical content.

Method	Type	λ Sch.	Obj. Func.
Forward KL (Jerfel et al., 2021)	Forward	–	$\mathbb{E}_P[\log(P/Q)]$
Backward KL (Malinin & Gales, 2019)	Backward	–	$\mathbb{E}_Q[\log(Q/P)]$
Fixed-Param BiKL (Bai et al., 2024)	Bidirectional	Constant	$\lambda_{\text{fix}}(\text{KL}_\text{F} + \text{KL}_\text{R})$
BiKL (Li et al., 2024)	Bidirectional	λ=1	$\text{KL}_\text{F} + \text{KL}_\text{R}$
Stepped BiKL (Yang et al., 2025)	Bidirectional	Adaptive	$\lambda(t)\text{KL}_\text{F} + [1\text{-}\lambda(t)]\text{KL}_\text{R}$

Table 2: KL Divergence Method. a) Forward KL: Forward KL Divergence, b) Backward KL: Backward KL Divergence, Fixed-Param BiKL: Fixed-Parameter Bi-directional KL Divergence, BiKL: Bi-directional KL Divergence, and Stepped BiKL: Stepped Bi-directional KL Divergence. Specifically, Forward and Backward KL Divergence only have one direction. Bi-directional KL has both forward and backward, yet it realizes this through three distinct methods, as presented in the table (where the formulas in the "Obj. Func." column correspond to these different realization logics)

4.2 Experimental Setup

All experiments to train the model are performed on a hardware environment equipped with an RTX 4090 GPU (24GB), a 16 vCPU Intel(R) Xeon(R) Gold 6430 CPU, Pytorch version 2.5.1, and the operating system is Ubuntu 18.04.

4.3 Model Training

For knowledge distillation, we used GPT-4o to generate 200 music-related texts as prompts that guide the logits generation of both student and teacher models, and we separate the datasets into train, validation, and test.

These prompts are meticulously crafted and include multi-dimensional musical attributes, such as temporal diversity, Genre, and instrumentation, and emotional and semantic nuances. MusicGen-small (the student model) was trained on GPU for approximately 30 hours for 10 epochs. To better show the performance of our loss function, we also trained the model when the loss function was set as Stage-mixed Bidirectional and Skewed KL Divergence, and other KL Divergence methods (as shown in Table 2), and we compare the loss score of these functions.

To perform the end-to-end conversion from PyTorch to ONNX format, after the knowledge distillation, we use the Optimum-Cli tool [2], for model optimization and deployment. This tool is chosen due to its native support for mixed-precision workflows and automated graph optimization, ensuring compatibility with downstream inference engines. We compare the performance of the various formats, include the original Torch version and the ONNX version with different quantization.

4.4 Mobile Device Deployment

After our model has been built, to deploy it on edge devices, we convert the model format from PyTorch to Open Neural Network Exchange (ONNX)[3]. ONNX is an open-source format for neural network models (Shridhar et al., 2020). To evaluate the model's performance in real-world scenarios, we deploy the converted ONNX model to on-device testing environments. The device we chose is the iPhone 16 Pro, operating on iOS 18.2, which is equipped with an A18 Pro processor with a 6-core GPU, and 8GB of RAM. We use ONNXRuntime to execute the ONNX model on edge devices. The running screenshot and codes are displayed in the appendix A.

4.5 Ablation Study

To investigate the individual and combined effects of knowledge distillation and quantization, we design four configurations based on MusicGen-Small, all evaluated on the MusicCap dataset (Lee et al., 2023) with consistent metrics as specified in subsection:

Baseline: Original MusicGen-Small model without knowledge distillation or quantization, maintaining the default architecture and parameters of the base model.

[2]https://github.com/huggingface/optimum

[3]https://github.com/onnx/models

MusicGen Small with KD: MusicGen-Small integrated with Stage-mixed Bidirectional KL-Divergence distillation (using MusicGen-Large as the teacher model) but without quantization, focusing on the impact of knowledge transfer alone.

MusicGen Small (Quantization): Original MusicGen-Small applied with adaptive mixed-precision quantization (targeting different components like Text-Encoder and MusicGen-Decoder) but without knowledge distillation, isolating the effect of quantization on efficiency and quality.

TinyMusician (KD + Quantization): Full TinyMusician framework, combining both Stage-mixed Bidirectional KL-Divergence distillation and adaptive mixed-precision quantization to evaluate the synergistic effect of the two techniques.

4.6 Comparison with State-of-the-Arts

We evaluate TinyMusician's performance across different formats and state-of-the-art AI music generation models, including YuE (Yuan et al., 2025), DiffRhythm (Ning et al., 2025), InspireMusic (Zhang et al., 2025), CRFM (Thickstun et al., 2023), Magenta-Realtime (Team, 2025), Musicgen-Small (Copet et al., 2023).

In particular, we incorporate resource utilization metrics and accuracy-related established benchmarks. The resource utilization includes inference time (in seconds), GPU FLOPS, CPU utilization (in percentage), Memory Usage (in GB), GPU Memory Usage (in GB), and model size (in gigabytes), which collectively characterize the models' computational efficiency and resource requirements.

To measure accuracy-related benchmarks, we employ two important benchmarks, i.e., CLAP (Contrastive Language Audio Pretraining) (Ye et al., 2023) and FADscore (Frechet Audio Distance Score) (Kilgour et al., 2018). CLAP is a framework that can capture the semantic relationship between audio and text. FADscore, on the other hand, measures the similarity between the distribution of generated audio and the real-world audio distribution.

5 Results and Discussion

In this section, we present and analyze the experimental results to validate the effectiveness of TinyMusician.

We first examine the training dynamics of different loss functions to highlight the advantages of our proposed *Stage-mixed Bidirectional KL divergence*. Next, we report the findings of the ablation study, which quantifies the individual and combined impacts of knowledge distillation and quantization on model performance and efficiency. Finally, we conduct a detailed comparison with state-of-the-art music generation systems, demonstrating the superior trade-off between efficiency and generation quality achieved by TinyMusician.

5.1 Training Dynamics of Loss Functions

The results presented in Figure 2 show that our proposed *Stage-mixed Bidirectional KL divergence* loss function stabilizes training dynamics and enhances model generalization.

The superior performance of our method is characterized by smooth training loss descent, minimal validation loss fluctuations, and the lowest final loss (≈ 0.13). Unlike single-directional KL divergence (Forward/Backward), which suffers from late-stage instability, our bidirectional formulation balances these two directions through dynamic weighting ($\alpha(t)$).

Furthermore, our method avoids the dramatic oscillations of **Fixed bi-KL divergence** and **Baseline Bi-KL divergence**, highlighting its strong generalization ability.

This is particularly valuable for music generation tasks, as overfitting to training data (indicated by volatile validation loss) would lead to inconsistent audio quality, such as abrupt shifts in melody or rhythm. Lastly, even though Stepped Bi-KL shows a similar pattern and performance, our method still demonstrates better results in the late stages. The results shown in Figure 3 (b) present the superiority of our method. Among all compared KL Divergence formulations, our bidirectional KL

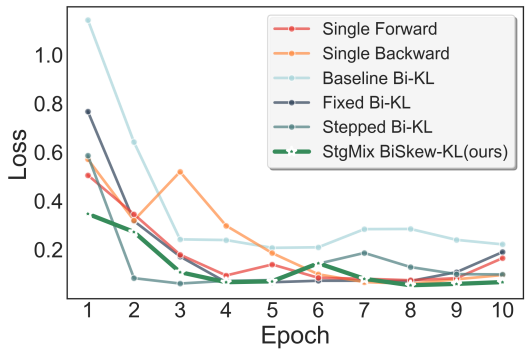


Figure 2: Comparison of Training Loss on different KL-Divergence Methods

Divergence with dynamic weighting achieves the lowest test loss. This not only indicates more stable training convergence but also reflects stronger generalization capability.

This performance originates from two synergistic mechanisms: (i) In the early training stages, the model prioritizes learning the teacher's overall structural patterns by emphasizing the divergence from the teacher to the student's smoothed output. As training progresses beyond a predefined threshold (τ_{step}), the focus shifts to refining local temporal details critical for music—such as rhythmic consistency and melodic flow—by emphasizing the divergence from the student to the teacher's adjusted output. This stage-specific adaptation of weight distribution ($\alpha(t)$) is governed by a dynamic coefficient that transitions from 1 to 0 at τ_{step}, ensuring the model balances global pattern learning and local detail preservation during different training phases.

(ii) The use of blended distributions (combining teacher and student outputs) prevents the student from overfitting to the teacher's specific non-generalizable patterns, while creating a stable reference frame for capturing long-range musical dependencies like harmonic progressions or thematic repetitions.

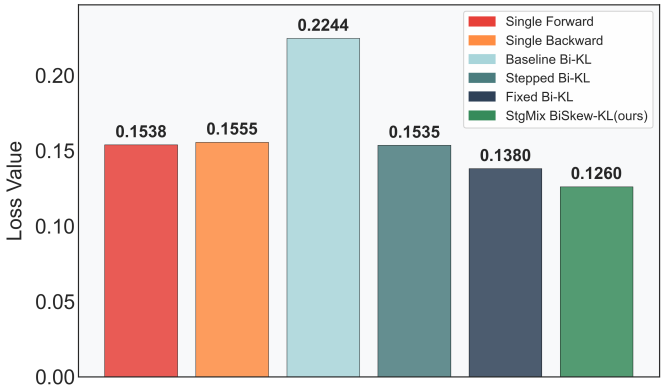


Figure 3: Comparison of Test Loss

5.2 Ablation Study

To evaluate how Knowledge Distillation (KD) and Quantization shape the performance and efficiency of TinyMusician among MusicGen-Small, we conduct an ablation study by isolating their individual impacts and analyzing their combined effect.

5.2.1 Impact of Knowledge Distillation

Table 3 shows the result of accuracy. KD marginally improves generation quality (FAD score drops from 6.49 to 6.44) but slightly degrades text-audio alignment (CLAP score falls from 0.303 to

Model	FADscore ↓	CLAPscore ↑
MusicGen-Small (Baseline)	6.49	0.303
TinyMusician	**6.44**	0.301
MusicGen-Small + Quantization	7.11	**0.352**
TinyMusician + Quantization	7.05	0.343

Table 3: The Scores of Ablation Study. FADscore ↓ represents lower is better, CLAPscore ↑ represents higher is better

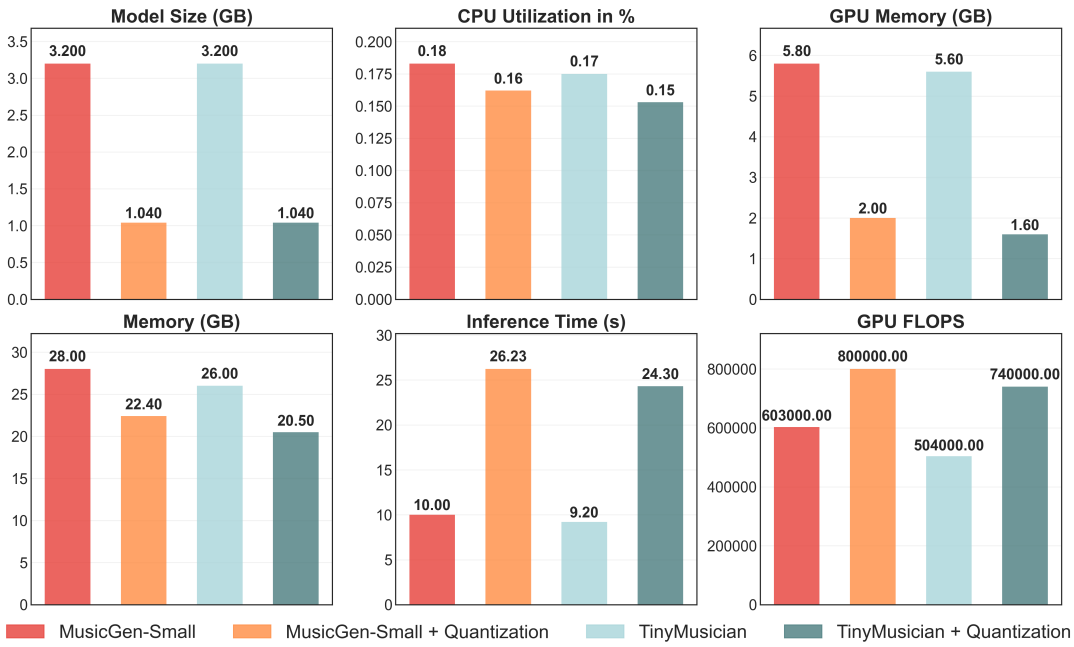


Figure 4: Resource Utilization Comparison between TinyMusician and MusicGen-Small with different configurations.

0.301). This suggests KD preserves fine-grained audio details from the teacher model but may dilute text-guided conditioning. Notably, KD alone does not change the model's efficiency in traditional metrics: our measurements (consistent with Figure 4) show that compared to the baseline MusicGen-Small, the TinyMusician with KD-optimized retains nearly identical inference latency and memory footprint. This stability in efficiency metrics is likely due to the preservation of the baseline's architectural backbone during distillation, where knowledge transfer focuses on refining output quality rather than reducing model size or computational complexity.

5.2.2 Impact of Quantization

As shown in Table 3, quantization drastically boosts text-audio alignment (CLAP score jumps to 0.352) but harms generation quality (FAD score rises to 7.11). This trade-off is initially counterintuitive, as quantization typically reduces model precision. Quantization acts as a form of regularization (Moradi et al., 2020), constraining model complexity and reducing overfitting (Ying, 2019) on the alignment task. When combined, KD and quantization strike a balance: the joint approach achieves a FAD score of 7.05 (better than quantization alone) and retains strong alignment (CLAP score 0.343). As shown in Figure 4, quantization delivers extreme compression (model size shrinks from 3.2GB to 1.04GB, GPU memory from 5.8GB to 2GB), while KD adds negligible overhead. However, the joint method inherits quantization's latency penalty (26.54s vs. 10s for the baseline), likely due to unoptimized quantized kernels on our test hardware. These results underscore the potential of hybrid optimization strategies, contingent on hardware-aware deployment.

Model	FADscore ↓	CLAPscore ↑
CRFM	8.9	0.222
InspireMusic-Base	10.7	0.150
YuE-7B	10.41	0.310
DiffRhythm	10.97	0.16
Mageneta-Realtime	**6.79**	0.311
MusicGen-Small (Baseline)	6.49	0.303
TinyMusician	6.44	0.301
TinyMusician-Int8	8.30	0.283
TinyMusician-MixedPrecision	7.05	**0.373**

Table 4: TinyMusician model performance in comparison to state-of-the-art models.

5.3 Comparison with State-of-the-Art Models

Table 4 and Figure 5 reflect the performance of models across various dimensions, including model size, resource consumption (CPU/GPU utilization, memory), inference efficiency (time, FLOPs), and quality metrics (FADscore (Kilgour et al., 2018), CLAPscore (Ye et al., 2023)). Focusing on the state-of-the-art models, MusicGen-Small quantization variants, the MusicGen-Small/ONNX(KD) Mixed configuration demonstrates outstanding trade-off advantages.

Large-scale models (e.g., YuE-7B, DiffRhythm) prioritize fidelity but suffer from prohibitive size and latency, while pure Int-8 quantization (0.58GB) sacrifices musical coherence (e.g., abrupt rhythm shifts). In contrast, the TinyMusician-MixedPrecision approach strikes a critical balance: its 1.04GB footprint retains near-baseline fidelity with FADscore of 7.05, approaching full-precision's 6.44, and outperforms all competitors with CLAPscore of 0.373, underscoring stronger text-music alignment. Though its 26.54s latency marginally exceeds INT8, it remains orders of magnitude faster than bulky models (e.g., YuE-7B's 1007s). The MusicGen-Tiny variant further validates scalability: with 40% fewer parameters, it nears baseline fidelity, demonstrating the framework's potential for efficiency-driven refinement.

This hierarchy highlights a core insight: naive compression (Int-8) or scale (large models) fails to serve on-device music generation, whereas our TinyMusician-MixedPrecision strategy harmonizes compactness with perceptual quality — a critical requirement for edge deployment.

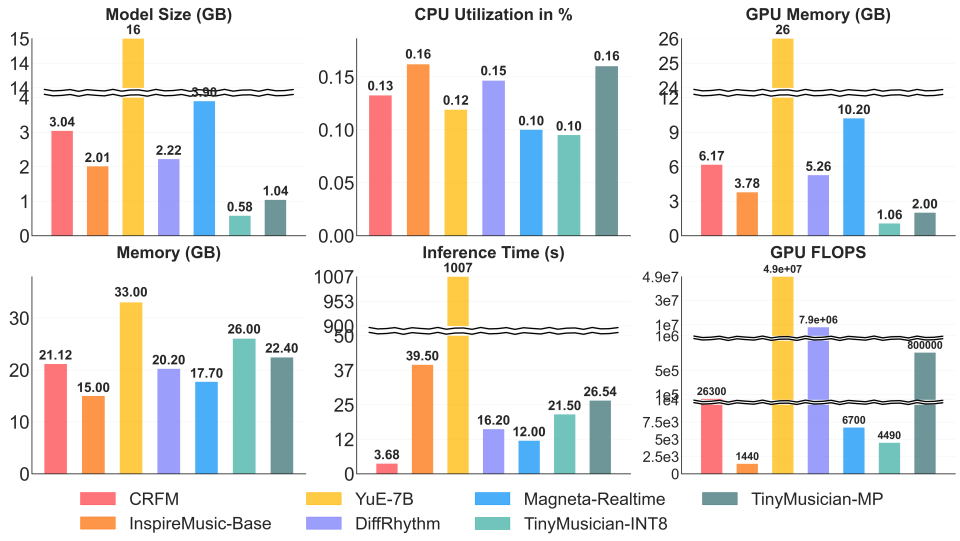


Figure 5: Resource utilization comparison among different models. TinyMusician-MP: TinyMusician-MixedPrecision.

6 CONCLUSION

In this study, we address the critical challenge of deploying large music generation models on resource-constrained edge devices, such as mobile phones, by introducing TinyMusician, a lightweight framework that integrates knowledge distillation and adaptive mixed-precision quantization. By using MusicGen as a baseline, we propose a stage-mixed bidirectional KL Divergence loss with a dynamic temperature annealing strategy to enhance the performance of knowledge transfer between the teacher and student models. To further optimize inference efficiency on devices, we apply mixed-precision quantization to different components of the MusicGen model, achieving a 55% reduction in model size while preserving the performance. For future work, we plan to investigate how to speed up on-device inference by using the different model formats. Also, could apply this framework to the state-of-the-art generative models and conduct more experiments to optimize further compression strategies while saving the output quality.

REFERENCES

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

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Yanfeng Bai, Haitao Wang, and Jianfeng He. Blin: A multi-task sequence recommendation based on bidirectional kl-divergence and linear attention. *Mathematics*, 12(15):2391, 2024.

Leonardo Berti, Flavio Giorgi, and Gjergji Kasneci. Emergent abilities in large language models: A survey. *arXiv preprint arXiv:2503.05788*, 2025.

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, et al. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*, 2024.

Arun Chauhan, Utsav Tiwari, et al. Post training mixed precision quantization of neural networks using first-order information. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1343–1352, 2023.

Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez. Simple and controllable music generation. *Advances in Neural Information Processing Systems*, 36:47704–47720, 2023.

Adyasha Dash and Kathleen Agres. Ai-based affective music generation systems: A review of methods and challenges. *ACM Computing Surveys*, 56(11):1–34, 2024.

Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression. *arXiv preprint arXiv:2210.13438*, 2022.

Olivier Derrien, Pierre Duhamel, Maurice Charbit, and Gaël Richard. A new quantization optimization algorithm for the mpeg advanced audio coder using a statistical subband model of the quantization noise. *IEEE transactions on audio, speech, and language processing*, 14(4):1328–1339, 2006.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341*, 2020.

Mohit Dua, Rohit Yadav, Divya Mamgai, and Sonali Brodiya. An improved rnn-lstm based novel approach for sheet music generation. *Procedia Computer Science*, 171:465–474, 2020.

Steven K Esser, Jeffrey L McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S Modha. Learned step size quantization. *arXiv preprint arXiv:1902.08153*, 2019.

Zhengcong Fei, Mingyuan Fan, Changqian Yu, and Junshi Huang. Flux that plays music. *arXiv preprint arXiv:2409.00587*, 2024.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*, 2022.

Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. A survey of quantization methods for efficient neural network inference. In *Low-power computer vision*, pp. 291–326. Chapman and Hall/CRC, 2022.

Kratarth Goel, Raunaq Vohra, and Jajati Keshari Sahoo. Polyphonic music generation by modeling temporal dependencies using a rnn-dbn. In *Artificial Neural Networks and Machine Learning–ICANN 2014: 24th International Conference on Artificial Neural Networks, Hamburg, Germany, September 15-19, 2014. Proceedings 24*, pp. 217–224. Springer, 2014.

Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819, 2021.

Zaky Hassani and Aciek Ida Wuryandari. Music generator with markov chain: A case study with beatme touchdown. In *2016 6th international conference on system engineering and technology (ICSET)*, pp. 179–183. IEEE, 2016.

Khafiizh Hastuti, A Azhari, A Musdholifah, and R Supanggah. Rule-based and genetic algorithm for automatic gamelan music composition. *International Review on Modelling and Simulations*, 10(3):202–212, 2017.

Curtis Hawthorne, Ian Simon, Adam Roberts, Neil Zeghidour, Josh Gardner, Ethan Manilow, and Jesse Engel. Multi-instrument music synthesis with spectrogram diffusion. *arXiv preprint arXiv:2206.05408*, 2022.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Lu Wangpit Wang, and Trevor Neel. LoRA: Low-Rank Adaptation of Large Language Models. *arXiv*, 2021. URL `https://arxiv.org/abs/2106.09685`.

Chih-Fang Huang and Cheng-Yuan Huang. Emotion-based ai music generation system with cvae-gan. In *2020 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE)*, pp. 220–222. IEEE, 2020.

Yudong Huang, Hongyang Du, Xinyuan Zhang, Dusit Niyato, Jiawen Kang, Zehui Xiong, Shuo Wang, and Tao Huang. Large language models for networking: Applications, enabling techniques, and challenges. *IEEE Network*, 2024.

Nabil Ibtehaz and M Sohel Rahman. Multiresunet: Rethinking the u-net architecture for multimodal biomedical image segmentation. *Neural networks*, 121:74–87, 2020.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive Mixtures of Local Experts. *Neural Computation*, 3(1):79–87, 1991.

Ghassen Jerfel, Serena Wang, Clara Wong-Fannjiang, Katherine A Heller, Yian Ma, and Michael I Jordan. Variational refinement for importance sampling using the forward kullback-leibler divergence. In *Uncertainty in Artificial Intelligence*, pp. 1819–1829. PMLR, 2021.

Maximos Kaliakatsos-Papakostas, Andreas Floros, and Michael N Vrahatis. Artificial intelligence methods for music generation: a review and future perspectives. *Nature-inspired computation and swarm intelligence*, pp. 217–245, 2020.

Jaeyong Kang, Soujanya Poria, and Dorien Herremans. Video2music: Suitable music generation from videos using an affective multimodal transformer model. *Expert Systems with Applications*, 249:123640, 2024.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Kevin Kilgour, Mauricio Zuluaga, Dominik Roblek, and Matthew Sharifi. Fr\'echet audio distance: A metric for evaluating music enhancement algorithms. *arXiv preprint arXiv:1812.08466*, 2018.

Max WY Lam, Qiao Tian, Tang Li, Zongyu Yin, Siyuan Feng, Ming Tu, Yuliang Ji, Rui Xia, Mingbo Ma, Xuchen Song, et al. Efficient neural music generation. *Advances in Neural Information Processing Systems*, 36:17450–17463, 2023.

Minhee Lee, Seungheon Doh, and Dasaem Jeong. Annotator subjectivity in the musiccaps dataset. *HCMIR@ ISMIR*, 8, 2023.

Li Li, Yuxi Fan, Mike Tse, and Kuo-Yi Lin. A review of applications in federated learning. *Computers & Industrial Engineering*, 149:106854, 2020.

Minchong Li, Feng Zhou, and Xiaohui Song. Bild: Bi-directional logits difference loss for large language model distillation. *arXiv preprint arXiv:2406.13555*, 2024.

Xia Liang, Junmin Wu, and Jing Cao. Midi-sandwich2: Rnn-based hierarchical multi-modal fusion generation vae networks for multi-track symbolic music generation. *arXiv preprint arXiv:1909.03522*, 2019.

Lei Liu, Xiaoyan Yang, Yue Shen, Binbin Hu, Zhiqiang Zhang, Jinjie Gu, and Guannan Zhang. Think-in-memory: Recalling and post-thinking enable llms with long-term memory. *arXiv preprint arXiv:2311.08719*, 2023.

Mingyang Liu. Overview of artificial intelligence painting development and some related model application. In *SHS Web of Conferences*, volume 167, pp. 01004. EDP Sciences, 2023.

Ruikang Liu, Yuxuan Sun, Manyi Zhang, Haoli Bai, Xianzhi Yu, Tiezheng Yu, Chun Yuan, and Lu Hou. Quantization hurts reasoning? an empirical study on quantized reasoning models. *arXiv preprint arXiv:2504.04823*, 2025.

Debasmita Lohar, Clothilde Jeangoudoux, Anastasia Volkova, and Eva Darulova. Sound mixed fixed-point quantization of neural networks. *ACM Transactions on Embedded Computing Systems*, 22(5s):1–26, 2023.

Roisin Loughran and Michael O'Neill. Evolutionary music: applying evolutionary computation to the art of creating music. *Genetic Programming and Evolvable Machines*, 21:55–85, 2020.

Heng Lu, Mehdi Alemi, and Reza Rawassizadeh. The impact of quantization and pruning on deep reinforcement learning models. *arXiv preprint arXiv:2407.04803*, 2024.

Andrey Malinin and Mark Gales. Reverse kl-divergence training of prior networks: Improved uncertainty and adversarial robustness. *Advances in neural information processing systems*, 32, 2019.

Rohin Manvi, Anikait Singh, and Stefano Ermon. Adaptive inference-time compute: Llms can predict if they can do better, even mid-generation. *arXiv preprint arXiv:2410.02725*, 2024.

Jan Melechovsky, Zixun Guo, Deepanway Ghosal, Navonil Majumder, Dorien Herremans, and Soujanya Poria. Mustango: Toward controllable text-to-music generation. *arXiv preprint arXiv:2311.08355*, 2023.

Reza Moradi, Reza Berangi, and Behrouz Minaei. A survey of regularization strategies for deep models. *Artificial Intelligence Review*, 53(6):3947–3986, 2020.

Ravi Teja Mullapudi, Steven Chen, Keyi Zhang, Deva Ramanan, and Kayvon Fatahalian. Online model distillation for efficient video inference. In *Proceedings of the IEEE/CVF International conference on computer vision*, pp. 3573–3582, 2019.

Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith B Hall, Daniel Cer, and Yinfei Yang. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. *arXiv preprint arXiv:2108.08877*, 2021.

Ziqian Ning, Huakang Chen, Yuepeng Jiang, Chunbo Hao, Guobin Ma, Shuai Wang, Jixun Yao, and Lei Xie. Diffrhythm: Blazingly fast and embarrassingly simple end-to-end full-length song generation with latent diffusion. *arXiv preprint arXiv:2503.01183*, 2025.

Reza Rawassizadeh. *Machine Learning and Artificial Intelligence: Concepts, Algorithms and Models*. Reza Rawassizadeh, 2025.

Reza Rawassizadeh and Yi Rong. ODSearch: Fast and Resource Efficient On-device Natural Language Search for Fitness Trackers' Data. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6(4):1–25, 2023.

Reza Rawassizadeh, Timothy J Pierson, Ronald Peterson, and David Kotz. NoCloud: Exploring network disconnection through on-device data analysis. *IEEE Pervasive Computing*, 17(1):64–74, 2018.

Babak Rokh, Ali Azarpeyvand, and Alireza Khanteymoori. A comprehensive survey on model quantization for deep neural networks in image classification. *ACM Transactions on Intelligent Systems and Technology*, 14(6):1–50, 2023.

Guillaume Sanchez, Honglu Fan, Alexander Spangher, Elad Levi, Pawan Sasanka Ammanamanchi, and Stella Biderman. Stay on topic with classifier-free guidance. *arXiv preprint arXiv:2306.17806*, 2023.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

Florian Schmid, Khaled Koutini, and Gerhard Widmer. Efficient large-scale audio tagging via transformer-to-cnn knowledge distillation. In *ICASSP 2023-2023 IEEE international Conference on acoustics, Speech and signal processing (ICASSP)*, pp. 1–5. IEEE, 2023.

Yuzhang Shang, Zhihang Yuan, Bin Xie, Bingzhe Wu, and Yan Yan. Post-training quantization on diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1972–1981, 2023.

Ilana Shapiro and Mark Huber. Markov chains for computer music generation. *Journal of humanistic mathematics*, 11(2):167–195, 2021.

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

Ayush Shridhar, Phil Tomson, and Mike Innes. Interoperating deep learning models with onnx. jl. In *Proceedings of the JuliaCon Conferences*, volume 1, pp. 59, 2020.

Carmen Silva and Lídia Oliveira. Artificial intelligence at the interface between cultural heritage and photography: A systematic literature review. *Heritage*, 7(7):3799–3820, 2024.

Jon Sneyers and Danny De Schreye. Apopcaleaps: Automatic music generation with chrism. In *Proceedings of 22nd Benelux Conference on Artificial Intelligence (BNAIC'10)*, pp. 1–8, 2010.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Peiming Sun. A study of artificial intelligence in the production of film. In *SHS Web of Conferences*, volume 183, pp. 03004. EDP Sciences, 2024.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. Patient knowledge distillation for bert model compression. *arXiv preprint arXiv:1908.09355*, 2019a.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. Patient knowledge distillation for bert model compression. *arXiv preprint arXiv:1908.09355*, 2019b.

Suno-Ai. Suno-AI: text-prompted Generative Audio Model, May 2023. URL https://github.com/suno-ai/bark.

Yehui Tang, Yunhe Wang, Jianyuan Guo, Zhijun Tu, Kai Han, Hailin Hu, and Dacheng Tao. A survey on transformer compression. *arXiv preprint arXiv:2402.05964*, 2024.

Lyria Team. Magenta realtime. 2025. URL https://g.co/magenta/rt.

John Thickstun, David Hall, Chris Donahue, and Percy Liang. Anticipatory music transformer. *arXiv preprint arXiv:2306.08620*, 2023.

Qi Tian, Jianxiao Zou, Jianxiong Tang, Yuan Fang, Zhongli Yu, and Shicai Fan. Mrcnn: a deep learning model for regression of genome-wide dna methylation. *BMC genomics*, 20:1–10, 2019.

Zhengyu Tian, Anantha Padmanaban Krishna Kumar, Hemant Krishnakumar, and Reza Rawassizadeh. Attentions Under the Microscope: A Comparative Study of Resource Utilization for Variants of Self-Attention. *arXiv preprint arXiv:2507.07247*, 2025.

Jason Toynbee. Music, culture, and creativity. In *The cultural study of music*, pp. 161–171. Routledge, 2012.

Lu Wei, Zhong Ma, Chaojie Yang, and Qin Yao. Advances in the neural network quantization: A comprehensive review. *Applied Sciences*, 14(17):7445, 2024.

Qingsong Wen, Tian Zhou, Chaoli Zhang, Weiqi Chen, Ziqing Ma, Junchi Yan, and Liang Sun. Transformers in time series: A survey. *arXiv preprint arXiv:2202.07125*, 2022.

Chuhan Wu, Fangzhao Wu, and Yongfeng Huang. One teacher is enough? pre-trained language model distillation from multiple teachers. *arXiv preprint arXiv:2106.01023*, 2021.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. Smoothquant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning*, pp. 38087–38099. PMLR, 2023.

Menghao Yang, Yafei Qi, Zhaoning Zhang, Yaping Liu, and Bing Xiong. Modal mimicking knowledge distillation for monocular 3d object detection. *Available at SSRN 5219975*, 2025.

Yao Yao, Peike Li, Boyu Chen, and Alex Wang. Jen-1 composer: A unified framework for high-fidelity multi-track music generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 14459–14467, 2025.

Zhenhui Ye, Rongjie Huang, Yi Ren, Ziyue Jiang, Jinglin Liu, Jinzheng He, Xiang Yin, and Zhou Zhao. Clapspeech: Learning prosody from text context with contrastive language-audio pre-training. *arXiv preprint arXiv:2305.10763*, 2023.

Xue Ying. An overview of overfitting and its solutions. In *Journal of physics: Conference series*, volume 1168, pp. 022022. IOP Publishing, 2019.

Ruibin Yuan, Hanfeng Lin, Shuyue Guo, Ge Zhang, Jiahao Pan, Yongyi Zang, Haohe Liu, Yiming Liang, Wenye Ma, Xingjian Du, et al. Yue: Scaling open foundation models for long-form music generation. *arXiv preprint arXiv:2503.08638*, 2025.

Chong Zhang, Yukun Ma, Qian Chen, Wen Wang, Shengkui Zhao, Zexu Pan, Hao Wang, Chongjia Ni, Trung Hieu Nguyen, Kun Zhou, Yidi Jiang, Chaohong Tan, Zhifu Gao, Zhihao Du, and Bin Ma. Inspiremusic: Integrating super resolution and large language model for high-fidelity long-form music generation. 2025. URL https://arxiv.org/abs/2503.00084.

Haohang Zhang, Letian Xie, and Kaiyi Qi. Implement music generation with gan: A systematic review. In *2021 International Conference on Computer Engineering and Application (ICCEA)*, pp. 352–355. IEEE, 2021.

Yihua Zhang, Yuguang Yao, Parikshit Ram, Pu Zhao, Tianlong Chen, Mingyi Hong, Yanzhi Wang, and Sijia Liu. Advancing model pruning via bi-level optimization. *Advances in Neural Information Processing Systems*, 35:18309–18326, 2022.

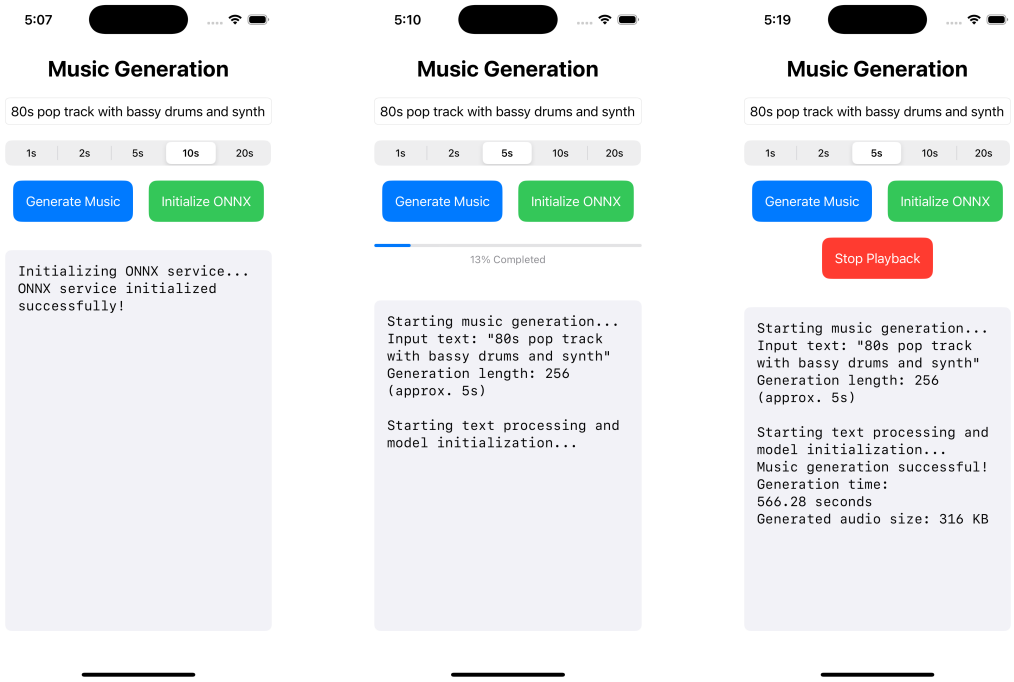
Zongmeng Zhang, Yufeng Shi, Jinhua Zhu, Wengang Zhou, Xiang Qi, Peng Zhang, and Houqiang Li. Trustworthy alignment of retrieval-augmented large language models via reinforcement learning. *arXiv preprint arXiv:2410.16843*, 2024.

Michael Zhu and Suyog Gupta. To prune, or not to prune: exploring the efficacy of pruning for model compression. *arXiv preprint arXiv:1710.01878*, 2017.

Pengfei Zhu, Chao Pang, Yekun Chai, Lei Li, Shuohuan Wang, Yu Sun, Hao Tian, and Hua Wu. Ernie-music: Text-to-waveform music generation with diffusion models. *arXiv preprint arXiv:2302.04456*, 2023.

A APPENDIX

We use Xcode to develop a music generation app and deploy an ONNX model on devices by using the ONNXRuntime package. The screen is shown in Figure 6. Figure 6a (a) presents the main screen of the app. Figure 6a (b) shows the music generation process, and (c) provides the details after generating.



(a) Main Screen (b) Generating (c) Generate Complete

Figure 6: iOS Music Generation App Overview