Making the Most of Limited Data: Score-Aware Training for Text-to-Music Generation

Yun-Chen Cheng, Tzu-Hung Huang, Chih-Pin Tan
Graduate Institute of Communication Engineering, National Taiwan University, Taipei, Taiwan
{hhhhaura, zionhuang1107, tanchihpin0517}@gmail.com

***Abstract*—State-of-the-art text-to-music generation systems rely on massive proprietary datasets and industrial-scale compute, making it impossible to disentangle architectural contributions from resource advantages. We propose *score-aware training*, which treats audio-caption alignment score as a direct supervision signal throughout the pipeline. Rather than discarding low-scoring segments, we repurpose them via a CLAP-conditioned Beta noise timestep schedule that routes them to high-noise training regimes, acting as an effective implicit regularizer. Complementarily, segment-level filtering removes the most misaligned examples, and a two-stage caption procedure bridges the distribution gap between verbose training captions and concise inference prompts. A REPA auxiliary loss further transfers structured semantic knowledge from pretrained CLAP and MuQ encoders without additional data. Our 450M-parameter FluxAudio-based system, submitted to the ICME 2026 ATTM Grand Challenge Efficiency Track, ranked 2nd across both tracks in the objective evaluation and 3rd in the Efficiency Track in the final MOS evaluation.**

***Index Terms*—text-to-music generation, flow matching, quality-aware training, representation alignment, diffusion transformer**

I. Introduction

Text-to-music generation (TTM) has become a cornerstone of modern AI-driven music creation, powering commercial applications that allow creators to produce music through natural language prompts alone. This shift toward text-driven interfaces reflects a broader democratization of music production, lowering the barrier for non-musicians to express musical ideas and enabling new forms of human-AI creative collaboration. Driven by advances in latent diffusion models, flow matching, and large-scale Transformer architectures, TTM systems such as MusicGen [1] and Stable Audio Open [2] have achieved remarkable musical quality and semantic controllability. Yet state-of-the-art models are predominantly trained on massive proprietary datasets and industrial-scale computational infrastructure, creating a significant barrier for academic researchers who wish to study or reproduce these systems, let alone pursue algorithmic breakthroughs. Without controlled access to comparable data and compute, it becomes impossible to disentangle whether performance gaps stem from architectural choices or simply from differences in training resources [3].

The ICME 2026 Academic Text-to-Music (ATTM) Grand Challenge [3] directly confronts this barrier by establishing a fair-play benchmark in which all participants must train generative models strictly from scratch on a standardized, CC-licensed subset of the MTG-Jamendo dataset [4]. This controlled setting removes data scale and proprietary resources as confounding factors, enabling direct and reproducible comparison of algorithmic design choices. We participate in the **Efficiency Track**, which further imposes a strict upper bound of 500M parameters on the core generative model.

This setup raises a fundamental question: *when data volume, data source, and model capacity are all fixed, what determines the ceiling of a model's performance?* We argue that the answer lies in *how effectively* that data is used during training. In practice, large-scale music datasets such as MTG-Jamendo exhibit substantial heterogeneity in audio-caption alignment score: even within a single track, different segments vary widely in how faithfully they correspond to the associated caption, motivating fine-grained, segment-level score management throughout the training pipeline. Furthermore, this heterogeneity creates a fundamental trade-off between training data volume and alignment score: treating all segments uniformly risks polluting the training signal with misaligned examples, while aggressively filtering for only the highest-scoring segments sacrifices substantial amounts of potentially useful musical content.

These observations motivate a data-centric approach organized around the principle of **score-aware training**. We propose four complementary components: (i) **segment-level CLAP-guided filtering** to remove the most misaligned audio-caption pairs at sub-track granularity; (ii) a **CLAP-conditioned Beta noise timestep schedule** that repurposes lower-scoring segments by routing them to the high-noise training regime, where coarse content is useful and misalignment is less harmful [5]; (iii) a **two-stage caption procedure** that fine-tunes on LLM-rewritten captions to bridge the verbose-training vs. concise-inference distribution gap; and (iv) a **REPA auxiliary loss** [6] that transfers structured semantic knowledge from pretrained CLAP [7] and MuQ [8] encoders without additional data.

Beyond data utilization, training a generative model from scratch on limited data poses a representation-learning challenge: the model must simultaneously discover the acoustic structure of music and align it with complex textual concepts. Yet structured semantic spaces capturing precisely these relationships have already been established by large-scale discriminative models. Rather than forcing our resource-constrained model to learn these abstractions in isolation, we incorporate a **representation alignment (REPA)** [6] auxiliary loss that aligns the model's internal representations with pre-

trained embeddings from CLAP [7] and MuQ [8], transferring structured audio-semantic knowledge without additional data. In ablation, this yields a $+0.018$ CLAP score improvement and reduces FAD from 0.2856 to 0.2767.

Together, these four components demonstrate that careful handling of data quality and training dynamics can substantially advance text-to-music generation in the absence of industrial-scale resources.

II. Methodology

Our approach is organized around a unifying principle of *score-aware training*: rather than treating all training data and all training steps equally, we systematically adapt each component of the pipeline—data selection, noise scheduling, text conditioning, and representation learning—based on an estimate of sample quality. We describe each component in turn below.

A. Segment Filtering Pipeline

The first step in our score-aware training pipeline is to ensure that only sufficiently well-aligned audio-caption pairs enter the training pool. We begin by analyzing the CLAP score distribution of the validation dataset, computed on one randomly sampled 10-second segment from each of the 1,000 validation audio files. As shown in Fig. 1, the distribution has a mean of approximately 0.33, with a substantial proportion of segments exhibiting low CLAP scores. Moreover, since musical content can vary substantially within a single track (e.g., across verses, choruses, and instrumental sections), segments within the same audio file are likely to exhibit high variance in their CLAP scores, motivating a segment-level filtering strategy rather than file-level selection.

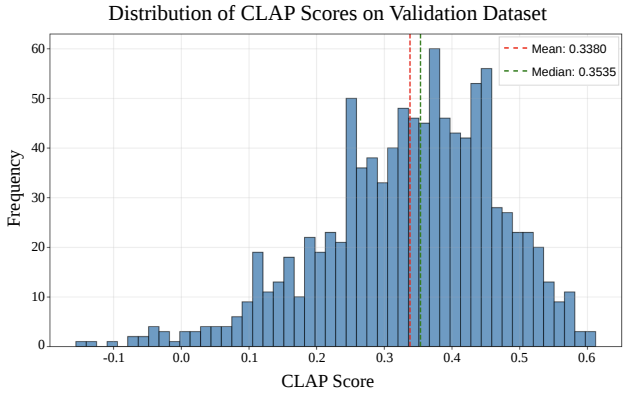

Fig. 1. CLAP score distribution of the validation dataset.

To ensure training data caption alignment quality, we apply a CLAP-guided segment selection pipeline. For each audio file, we randomly extract 15 candidate segments of 10 seconds each and compute the CLAP score for every segment. Segments are then partitioned into three quality tiers using two fixed thresholds: $\mathcal{S}_{\text{high}}$ (score $\geq$ 0.33), $\mathcal{S}_{\text{medium}}$ ($0.20 \leq$ score < 0.33), and $\mathcal{S}_{\text{low}}$ (score < 0.20). Low-score segments are discarded entirely.

We aim to retain exactly 6 segments per file from the high and medium tiers, following a priority scheme:

1) If the number of high-score segments $|\mathcal{S}_{\text{high}}| \geq 6$, we randomly sample 6 segments from $\mathcal{S}_{\text{high}}$.
2) Otherwise, we take all $|\mathcal{S}_{\text{high}}|$ high-score segments and fill the remaining $6 - |\mathcal{S}_{\text{high}}|$ slots with the top medium-score segments.

This strategy maximizes audio-caption alignment while maintaining sufficient training coverage per file.

B. Pretraining and Caption-Aligned Fine-tuning

We adopt a two-stage training procedure designed to first build broad musical knowledge from information-dense captions, and then specialize the model to the concise prompt style encountered at inference time.

a) Stage 1: Pretraining on Information-Dense Captions.: In the pretraining stage, we use both caption styles provided by the ATTM challenge. *Qwen-style* captions are generated directly by Qwen2-Audio-7B-Instruct [9], producing holistic descriptions of genre, instrumentation, and mood in a single pass. *MusicFlamingo-style* captions are generated by Music Flamingo [10] and subsequently refined by Qwen3-4B-Instruct into concise, natural-sounding descriptions. For each sample, one caption style is selected at random. Both styles are information-dense and frequently include fine-grained musical attributes such as tempo, key, time signature, and chord progressions, as illustrated by the following example:

> *"Built on C♯ major in 4/4 time at 120 BPM, it follows a loop-based structure with alternating chordal pads and melodic variations, including diatonic and chromatic shifts, a brief minor passage, and a resolved finale."*

This rich conditioning signal is well-suited for the early stages of training: it provides a dense, structured supervision target that encourages the model to learn fine-grained correspondences between textual attributes and acoustic content.

b) Stage 2: Fine-tuning on Inference-Style Captions.: While information-dense captions are valuable for pretraining, they introduce a distributional mismatch with the prompts encountered at inference time. Evaluation prompts tend to be concise and high-level, focusing on genre, instrumentation, and mood—for example:

> *"An upbeat EDM track with pulsing synths and driving bass."*

A model trained exclusively on dense captions may underperform when conditioned on these sparser prompts, simply because their distribution was never seen during training. To bridge this gap, we introduce a dedicated fine-tuning stage in which the model is adapted to the target inference-time caption distribution.

We construct the fine-tuning corpus by rewriting each caption into the target style with a large language model using the following prompt[1]:

[1]The `{examples}` placeholder is filled at runtime with a small set of few-shot demonstrations sampled from the target evaluation prompt distribution, anchoring the rewriter's output style to the desired inference-time format.

Caption Rewriting Prompt

```
You are a music caption editor. Rewrite
captions to include only:
1. Genre / style
2. Instrumentation (specific
instruments)
3. Mood, theme, or atmosphere

REMOVE: tempo/BPM, keys, time
signatures, chord progressions,
structural arcs, production/engineering
descriptions.

Write 1-2 natural flowing sentences.
No bullet points. Vary the openings.
Match this style: {examples}

Respond ONLY with a JSON array of
rewritten captions in the same order.
No preamble or markdown.
```

The rewriting step strips low-level musical attributes (tempo, key, chord progressions, structural arcs) while preserving the high-level semantic content (genre, instrumentation, mood). Fine-tuning on these rewritten captions teaches the model to ground generation in the same sparse, high-level descriptors that users provide at inference, without losing the fine-grained musical knowledge accumulated during pretraining.

C. CLAP-Conditioned Noise Timestep Scheduling

Rather than discarding segments whose alignment is imperfect but not negligible, our score-aware framework repurposes them by modulating their contribution to the training objective. Inspired by recent findings that low-quality data can still benefit generative training when used appropriately [5], we introduce a **CLAP-conditioned Beta noise timestep schedule** during flow matching. Instead of sampling the noise timestep $t \sim \mathcal{U}[0,1]$ uniformly for all samples, we condition the timestep distribution on the CLAP score $S \in [0,1]$ of each training segment using a Beta distribution:

$$P(t \mid S) = \mathrm{Beta}(t;\, \alpha(S),\, \beta(S))\,, \tag{1}$$

where $\beta(S) = 1$ and $\alpha(S)$ is a monotone function of S:

$$\alpha(S) = 1 + \lambda\,(1 - S), \quad \lambda = 1.0. \tag{2}$$

Here, S is the CLAP score normalized to $[0,1]$, and segments whose score exceeds the 75th percentile of the training distribution are treated as perfect quality, meaning their score is clipped to $S = 1$.

The intuition relies on the observation that flow-matching models learn at different fidelities across the noise trajectory. For a high-score segment ($S \approx 1$), we obtain $\alpha \approx 1$, recovering the uniform distribution $\mathrm{Beta}(1,1)$, which provides an equal training signal across all noise levels. Conversely, for a lower-score segment ($S \to 0$), we obtain $\alpha \approx 1 + \lambda$, which skews the Beta distribution toward $t \approx 1$ (the high-noise regime). At high t, predicting velocity only requires pointing in a roughly correct direction toward the clean data. By skewing the timestep sampling, lower-score segments contribute primarily to establishing this coarse semantic layout, without corrupting the fine-grained acoustic detail that is exclusively learned at low t. This targeted distribution of training signals is designed to act as an implicit regularizer, with the induced sampling behavior across score tiers summarized in Fig. 2.

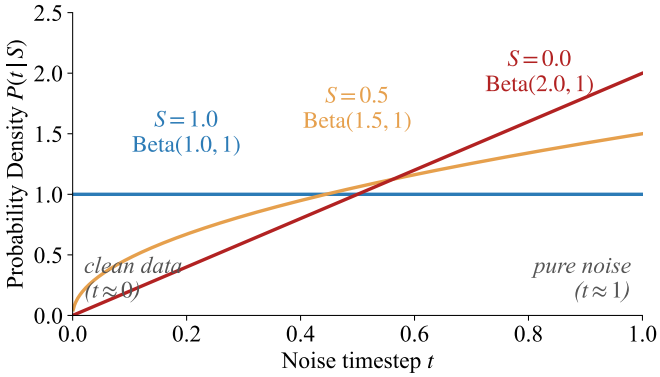

Fig. 2. Effect of the CLAP score S on the timestep sampling distribution under the proposed Beta schedule ($\lambda = 1.0$). High-score segments ($S = 1.0$) recover uniform sampling across all noise levels, while progressively lower-score segments concentrate their sampling mass near $t = 1$.

D. REPA Alignment Loss

a) Flow Matching Objective.: Our backbone is trained with a conditional flow matching objective. Given a clean audio latent $\mathbf{x}_1$ and a noise sample $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, the noisy latent at timestep $t \in [0,1]$ is constructed as $\mathbf{x}_t = (1-t)\,\mathbf{x}_1 + t\,\mathbf{x}_0$, and the model is trained to predict the target velocity $\mathbf{v} = \mathbf{x}_0 - \mathbf{x}_1$ via mean squared error:

$$\mathcal{L}_{\mathrm{FM}} = \left\| \hat{\mathbf{v}}_\theta(\mathbf{x}_t, t) - \mathbf{v} \right\|^2. \tag{3}$$

b) Adding Representation Alignment.: We augment this objective with a **Representation Alignment (REPA)** [6] auxiliary loss, which encourages the model's hidden states (after the joint and fused transformer blocks) to align with structured embeddings from pretrained semantic encoders. We instantiate two branches: one targeting global audio-text semantics, the other fine-grained musical structure.

For the **CLAP branch**, hidden states are mean-pooled and projected by a trainable head ϕ_{CLAP} into the CLAP embedding space, and the loss is the cosine distance to the CLAP embedding of the original audio:

$$\mathcal{L}_{\text{REPA-CLAP}} = 1 - \cos(\mathbf{z}_s,\, \mathbf{z}_{\mathrm{CLAP}})\,, \tag{4}$$

where $\mathbf{z}_s = \phi_{\mathrm{CLAP}}\left(\frac{1}{T}\sum_{n=1}^{T} \mathbf{h}_n\right) \in \mathbb{R}^d$, T denotes the audio sequence length (i.e., the number of frames rather than diffusion timesteps), $\mathbf{h}_n$ represents the hidden representation at the n-th frame, and $\mathbf{z}_{\mathrm{CLAP}} \in \mathbb{R}^d$ is the frozen CLAP embedding.

For the **MuQ branch** [8], which captures music-specific structure such as timbre and instrumentation that CLAP's

contrastive objective does not encode, we align at the *sequence level*. Since both student hidden states and MuQ features operate at 25 Hz, no resampling is needed: hidden states are projected frame-wise by a trainable head ϕ_{MuQ}, and the loss is the average cosine distance over all frames:

$$\mathcal{L}_{\text{REPA-MuQ}} = 1 - \frac{1}{T}\sum_{n=1}^{T} \cos(\mathbf{z}_{s,n},\ \mathbf{z}_{\text{MuQ},n})\,, \tag{5}$$

where $\mathbf{z}_{s,n} = \phi_{\text{MuQ}}(\mathbf{h}_n) \in \mathbb{R}^{d'}$ and $\mathbf{z}_{\text{MuQ},n} \in \mathbb{R}^{d'}$ is the corresponding frozen MuQ feature.

c) Timestep-Dependent Modulation.: Both losses are modulated by $w(t) = (1-t)^{\alpha}$ with $\alpha = 2.0$, weighting alignment more heavily at low noise levels where the latent closely resembles the original audio. The full objective is:

$$\begin{aligned}\mathcal{L} = \mathcal{L}_{\text{FM}} &+ \lambda_{\text{CLAP}} \cdot w_{\text{CLAP}}(t) \cdot \mathcal{L}_{\text{REPA-CLAP}} \\ &+ \lambda_{\text{MuQ}} \cdot w_{\text{MuQ}}(t) \cdot \mathcal{L}_{\text{REPA-MuQ}}.\end{aligned} \tag{6}$$

III. Experiment

A. Model Architecture

We adopt **FluxAudio** [11] as our flow matching backbone. FluxAudio is a FLUX-style [12] Diffusion Transformer (DiT) trained with a conditional flow matching objective on audio latents. For the latent representation consumed by the flow matching process, we use the pretrained **ACEStep 1.5** [13] audio codec as a frozen encoder, which encodes 48,000 Hz waveforms into continuous latent embeddings at 25 Hz. This provides compact, high-fidelity representations that preserve both acoustic detail and musical structure.

We condition the backbone on two complementary text representations: a **T5** [14] encoder provides fine-grained sequential token embeddings injected via cross-attention (sequence condition), while a **CLAP** [7] encoder provides a global semantic embedding applied through adaptive layer normalization (global condition). For the REPA alignment branches, hidden states are extracted after the final DiT block: the CLAP branch mean-pools and projects these into the CLAP embedding space, while the MuQ branch projects and injects them into layers 3 through 9 of the frozen MuQ conformer. The overall architecture is illustrated in Fig. 3.

B. Ablation Studies

To isolate the contribution of individual design choices, we conduct controlled ablations on a reduced-scale setting: 2,000 training samples, 100 validation samples, trained for 20,000 iterations using the smaller **FluxAudio-S** backbone (hidden dim 448, depth 12, fused depth 8, 7 attention heads). The base configuration disables all optional components; each ablation activates exactly one component at a time. We report **CLAP score** (audio-text alignment, ↑) and **FAD** over CLAP embeddings (↓) on the 100-sample validation set.

a) CLAP REPA: Both REPA configurations improve over the base, with the normal setting yielding the best audio-text alignment (+0.018 CLAP score). The aggressive setting achieves the lowest FAD at a minor cost to CLAP score.

b) MuQ REPA: The MuQ alignment run shows substantial degradation in both metrics. We attribute this to two compounding issues: (1) as shown in Fig. 4, validation loss has not converged within 20,000 iterations, indicating that MuQ alignment requires a longer training horizon; and (2) CLAP score is an insufficient metric for the music-specific structure that MuQ encodes—timbre, instrumentation, and key are not reflected in text-audio contrastive alignment.

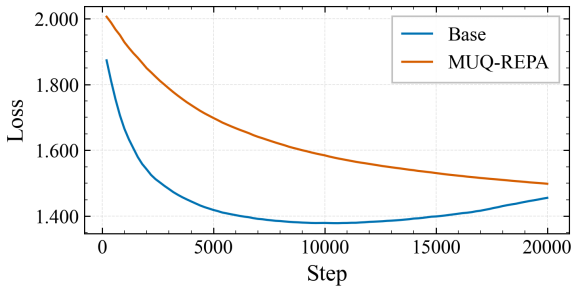

Fig. 4. Validation loss for base vs. MuQ REPA. The MuQ run starts from a higher initial loss (≈2.0 vs. ≈1.88) and converges substantially more slowly, confirming it has not converged within the 20,000-iteration budget.

c) Beta Noise Schedule: $\lambda = 0.2$ yields the best CLAP score (+0.003 over base), while $\lambda = 2.0$ degrades both metrics. The base achieves the best raw FAD, but this is misleading: as shown in Fig. 5, the base model overfits severely—validation loss reaches a minimum around step 7,500 before rising sharply to ≈1.46, while all Beta variants plateau at ≈1.35, a gap of 0.10 loss units.

Taken together, these ablations reveal a consistent theme: the primary challenge in this constrained setting is generalization rather than optimization. The base model overfits readily on the limited training set, as evidenced by the severe validation loss divergence in the Beta schedule ablation, demonstrating that the schedule acts as an effective implicit regularizer regardless of λ. CLAP REPA addresses this from a different angle, providing a semantically grounded auxiliary signal that stabilizes representation learning, yielding the most direct improvement in audio-text alignment (+0.018 CLAP score). The Beta noise schedule acts as a complementary implicit regularizer, reducing overfitting by approximately 0.10 validation loss units without sacrificing CLAP score. MuQ alignment, while theoretically well-motivated, requires a longer training horizon than our ablation budget permits and degrades both CLAP score and FAD within the 20,000-iteration budget.

IV. Submission Decision

A. Final Model Configuration

Based on our ablation studies, we configure the final submitted model with approximately **450M** trainable parameters. Table II summarizes the full architectural and training hyperparameters.

We retain **CLAP REPA** with the normal setting ($\alpha = 2.0$, $\lambda_{\text{CLAP}} = 0.2$), as it consistently improves both CLAP score and FAD in controlled ablations. For the **Beta noise schedule**,

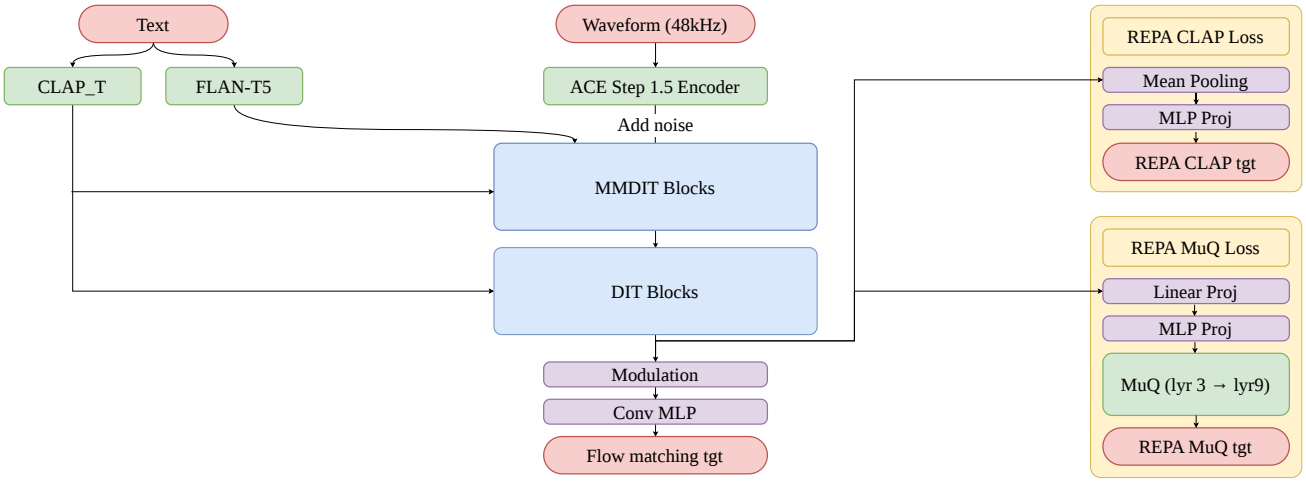

Fig. 3. Overview of our model architecture. The backbone is a FluxAudio Diffusion Transformer (DiT) conditioned on dual text representations: FLAN-T5 provides sequential token embeddings via cross-attention, while CLAP provides a global semantic embedding combined with the timestep embedding. Audio latents are encoded by the frozen ACEStep 1.5 codec. Hidden states extracted after the joint and fused transformer blocks are used for two auxiliary representation alignment branches: CLAP REPA (Setting 1 and 2) and MuQ REPA (Setting 2 only). At inference, the predicted flow velocity is decoded back to a waveform by the frozen ACEStep decoder.

TABLE I

UNIFIED ABLATION RESULTS. THE TOP ROW IS THE BASE CONFIGURATION WITH ALL OPTIONAL COMPONENTS DISABLED. EACH SUBSEQUENT GROUP ACTIVATES EXACTLY ONE COMPONENT WHILE INHERITING ALL OTHER BASE SETTINGS (SHOWN IN GRAY). ↑ HIGHER IS BETTER; ↓ LOWER IS BETTER.

Ablation on	**Configuration**	**CLAP REPA**	**MuQ REPA**	**Beta λ**	**CLAP ↑**	**FAD ↓**
Base	—	✗	✗	0	0.2755	0.2856
CLAP REPA	Normal (α=2.0, λ=0.2)	✓	✗	0	**0.2930**	0.2767
	Aggressive (α=4.0, λ=0.4)	✓	✗	0	0.2890	**0.2620**
MuQ REPA	$\alpha = 2.0, \lambda = 0.1$	✗	✓	0	0.1921	0.5864
Beta Schedule	$\lambda = 0.2$	✗	✗	0.2	**0.2788**	0.2941
	$\lambda = 1.0$	✗	✗	1.0	0.2746	0.2902
	$\lambda = 2.0$	✗	✗	2.0	0.2587	0.2995

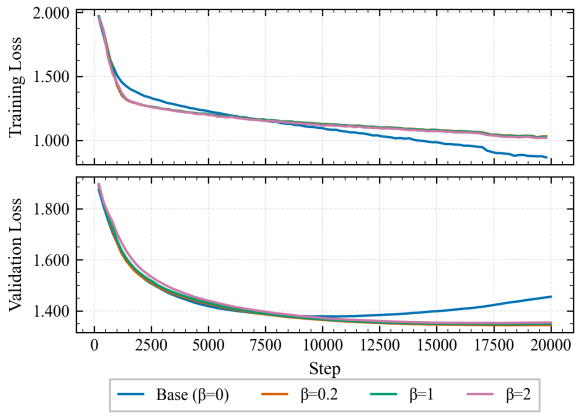

Fig. 5. Training (top) and validation (bottom) loss for the Beta schedule ablation. While the base model achieves the lowest training loss, it overfits severely after step ≈7,500, with validation loss rising sharply. All Beta variants cluster tightly in training and generalize substantially better.

we set $\lambda = 1.0$, which our ablations identify as stable and well-generalizing without over-aggressively suppressing training signal. We acknowledge that the ablations favor $\lambda = 0.2$; however, these results were finalized after the full training run had already been launched and could not be incorporated in time.

Given the uncertainty around MuQ alignment—which exhibited slow convergence within our ablation budget and could not be fairly assessed using CLAP-based metrics alone—we submit two variants: **Setting 1** (CLAP REPA + Beta) as our primary submission, and **Setting 2** (adds MuQ alignment) as an exploratory entry that did not fully converge before the deadline. Both settings undergo the caption rewrite fine-tuning stage described in Section II-B (10,000 additional steps on simplified captions), which improves CLAP score from 0.304 to 0.317 (+0.013) on the final submission prompt set.

B. Objective Phase Results

Our submission (Setting 1) achieves a CLAP score of 0.295, FAD of 0.495, and CCS of 0.804 on the final test prompts,

TABLE II
ARCHITECTURAL AND TRAINING HYPERPARAMETERS FOR THE FINAL SUBMITTED MODEL.

Parameter	Value
Architecture	
Latent dimension	64
Hidden dimension	896
Transformer depth	12
Fused depth	10
Number of attention heads	7
Latent sequence length	250
MLP ratio	4.0
Positional encoding	RoPE
REPA projection dim	512
MuQ projection dim	1,024
Total trainable parameters	≈450M
Training	
Learning rate	1×10^{-4}
Weight decay	1×10^{-6}
Gradient clipping	1.0
Linear warmup steps	1,000
LR schedule	Step ($\gamma = 0.1$)
Mixed precision (AMP)	Enabled
CLAP REPA	
λ_{CLAP}	0.2
α (timestep weight)	2.0
MuQ REPA	
Enabled	Setting 2 only
λ_{MuQ}	0.1
Beta Noise Schedule	
λ (Beta skew)	1.0
$\beta(S)$	1.0
75th-percentile clip	$S \leftarrow 1.0$
Finetuning on Inference Style Captions	
Steps	10,000
Caption subset	40% of training data

ranking **2nd across both tracks** in the objective evaluation phase and advancing as a finalist to the subsequent subjective listening test.

C. Human Evaluation Phase Results

Finalists advanced to a formal Mean Opinion Score (MOS) study conducted by expert listeners, evaluating *Audio Quality*, *Musicality*, and *Prompt Adherence*. Our system (e08) achieved $\text{MOS}_{\text{all}} = 3.119$ and $\text{MOS}_{\text{expert}} = 3.044$, placing **3rd in the Efficiency Track**.

V. CONCLUSION

We presented a score-aware training framework for text-to-music generation under the constraints of the ICME 2026 ATTM Challenge. By treating caption alignment score as a first-class signal informing filtering, noise scheduling, caption preparation, and representation learning, we showed that careful handling of training dynamics can advance TTM in academic settings without industrial-scale data or compute. Our final 450M-parameter system ranked 2nd across both tracks in the objective phase and 3rd in the Efficiency Track in the expert MOS evaluation. Ablations on a smaller-scale training subset indicated that CLAP REPA improved audio-text alignment ($+0.018$ CLAP score), the Beta noise schedule acted as a strong implicit regularizer (≈ 0.10 validation loss reduction), and caption rewriting helped bridge the training-inference distribution gap; these observations informed our final design, but full-scale verification is left to future work, along with optimizing the MuQ alignment objective over longer training horizons and extending the score-aware framework to other training signals beyond CLAP scores.

ACKNOWLEDGMENTS

The work is supported by grant from the Ministry of Education (MOE) of Taiwan (for Taiwan Centers of Excellence).

REFERENCES

[1] Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez, "Simple and controllable music generation," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[2] Zach Evans, Julian D Parker, CJ Carr, Zack Zukowski, Josiah Taylor, and Jordi Pons, "Stable audio open," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025.

[3] Fang-Chih Hsieh, Wei-Jaw Lee, Chun-Ping Wang, Hung-yi Lee, Hao-Wen Dong, and Yi-Hsuan Yang, "Academic text-to-music grand challenge: Datasets, baselines, and evaluation methods," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, 2026.

[4] Dmitry Bogdanov, Minz Won, Philip Tovstogan, Alastair Porter, and Xavier Serra, "The mtg-jamendo dataset for automatic music tagging," in *Machine Learning for Music Discovery Workshop, International Conference on Machine Learning (ICML 2019)*, Long Beach, CA, United States, 2019.

[5] Kenneth Li, Yida Chen, Fernanda Viégas, and Martin Wattenberg, "When bad data leads to good models," *arXiv preprint arXiv:2505.04741*, 2025.

[6] Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie, "Representation alignment for generation: Training diffusion transformers is easier than you think," in *International Conference on Learning Representations*, 2025.

[7] Benjamin Elizalde, Soham Deshmukh, Mahmoud Al Ismail, and Huaming Wang, "CLAP: Learning audio concepts from natural language supervision," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[8] Haina Zhu, Yizhi Zhou, Hangting Chen, Jianwei Yu, Ziyang Ma, Rongzhi Gu, Yi Luo, Wei Tan, and Xie Chen, "MuQ: Self-supervised music representation learning with mel residual vector quantization," *arXiv preprint arXiv:2501.01108*, 2025.

[9] Yunfei Chu et al., "Qwen-audio: Advancing universal audio understanding via unified large-scale audio-language models," *arXiv preprint arXiv:2311.07919*, 2023.

[10] Sreyan Ghosh et al., "Music flamingo: Scaling music understanding in audio language models," *arXiv preprint arXiv:2511.10289*, 2025.

[11] Xiquan Li, Junxi Liu, Yuzhe Liang, Zhikang Niu, Wenxi Chen, and Xie Chen, "Meanaudio: Fast and faithful text-to-audio generation with mean flows," *arXiv preprint arXiv:2508.06098*, 2025.

[12] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al., "Scaling rectified flow transformers for high-resolution image synthesis," in *Proceedings of the 41st International Conference on Machine Learning*, 2024.

[13] Junmin Gong, Yulin Song, Wenxiao Zhao, Sen Wang, Shengyuan Xu, and Jing Guo, "ACE-Step 1.5: Pushing the boundaries of open-source music generation," *arXiv preprint arXiv:2602.00744*, 2026.

[14] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," 2020, vol. 21, pp. 1–67.