Improving Text-to-Music Generation with Human Preference Rewards

Yonghyun Kim[1], Junwon Lee[2], Haiwen Xia[3], Yinghao Ma[4], Chris Donahue[5]
[1]Georgia Tech [2]KAIST [3]Peking University [4]Queen Mary University of London [5]Carnegie Mellon University

***Abstract*—We describe our entry to the efficiency track of the Academic Text-to-Music (ATTM) Grand Challenge at ICME 2026. Beyond the challenge protocol's FAD and CLAP-text metrics, we add a learned human-preference reward from *TuneJury*, a twin pairwise ranker trained over open music-preference datasets. The reward serves both as a training-time conditioning signal and as a sample-selection criterion. The pipeline combines five engineering decisions on a 120 M-parameter FluxAudio-S backbone, four at training time and one at inference: (i) training-time reward conditioning that doubles as an inference-time CFG axis, (ii) a sweep over five score-conditioning architectures, where training and inference use different variants, (iii) expert iteration on the top decile, (iv) a short preference-tuning pass (CRPO) for audio-caption alignment, and (v) inference post-processing via joint CFG, source separation, and loudness normalization. Per-stage decomposition on 100 Song Describer prompts shows training-time reward conditioning as a working conditioning axis, expert iteration as the dominant lift, the preference-tuning pass at noise level, and the inference-time score knob already saturated by the end of the chain.**

***Index Terms*—text-to-music generation, reward conditioning, preference modeling, flow matching, CFG, CRPO**

I. Introduction

This paper reports our submission to the efficiency track ($\leq$ 500 M parameters) of the Academic Text-to-Music (ATTM) Grand Challenge at ICME 2026 [1]. The challenge protocol evaluates three objective metrics: FAD-CLAP (Fréchet Audio Distance [2] on LAION-CLAP-Music audio embeddings [3]) against the SDD-706 reference, a 706-track instrumental subset of MTG-Jamendo [4] from the Song Describer Dataset (SDD) [5]; CLAP-text cosine similarity [3]; and a Concept Coverage Score (CCS) [1] computed by a large audio-language model judge. We focus on the first two in our internal tables and report the official CCS result in the Section IV footnote.

Beyond these two metrics, we use a learned human-preference reward supplied by *TuneJury* [6], a twin pairwise ranker [7] over LAION-CLAP-Music [3] and MERT [8] features, trained on open music-preference datasets. The reward enters the pipeline in two roles: a per-clip training-time conditioning signal, and a selection criterion for self-generated samples used in the expert-iteration fine-tune.

Our submission combines five concrete engineering decisions on the 120 M-parameter FluxAudio-S baseline [1], [9] provided by the challenge: four act on the backbone weights at training time, and one operates only at inference.

Training-time decisions.

Code & Demo: https://github.com/yonghyunk1m/ttm-humanpref.

(i) Conditioning on the human-preference reward. The per-clip TuneJury score enters the backbone as a Fourier-embedded [10] side input. Score-conditioned variants improve FAD-CLAP by 0.025–0.040 absolute over the unconditional baseline (Table I). Null-score dropout makes the reward an additional classifier-free guidance (CFG) [11] axis at inference time.

(ii) Sweep over score-conditioning heads. Of five injection heads on the Jamendo-100 validation set (Table I), InputAdd (v2) leads on FAD-CLAP, CLAP score, and input-score correlation. We deploy a v1 $\rightarrow$ v2 *hybrid*: train Stages 1–2 in the more-stable GlobalAdaLN (v1) forward, then cross-load into the InputAdd (v2) forward at Stage 3. The reverse cross is unsafe (Section IV-B).

(iii) Reward-guided expert iteration [12], [13]. We rank samples from the score-conditioned supervised fine-tuning (SFT) checkpoint by an equal-weight blend of ranker reward and CLAP-text similarity, and fine-tune on the top decile. This step is the dominant chain contributor: -0.0362 FAD-CLAP on the v1 chain (Row 1 $\rightarrow$ Row 2 of Table II).

(iv) Short preference tuning for CLAP-text alignment. A CLAP-Ranked Preference Optimization (CRPO) [14] pass with a direct preference optimization (DPO) [15]-style objective fine-tunes the expert-iteration checkpoint on $\sim$2K CLAP-aligned winner/loser pairs. The delta over expert iteration alone is within paired-t noise, but we keep the step because it is inexpensive.

Inference-time decision.

(v) Inference setup. Joint CFG [11] on text and reward, a fixed prompt prefix, 3$\times$Demucs [16] `mdx_extra` source separation, and LUFS normalization (Section III-E).

The remainder of the paper expands on these decisions (Section III) and reports per-stage ablations (Section IV).

a) Scope: The present report is scoped to the engineering pipeline of the submission: at inference, we use a fixed single-value score scalar selected on SDD-100, a 100-prompt held-out split of SDD (Section IV) and do not analyze the score-response curve. Three analytical questions are left to future work: (a) why reward-conditioned flow matching admits inference-time CFG *extrapolation* past the reward scalar's training support, (b) where this extrapolation breaks down, and (c) how it generalizes to other backbones.

b) Workflow note: The engineering reported here was carried out in a human-agent loop using Claude Code (Anthropic's Claude Opus 4.6/4.7), in the spirit of AI-Driven

Research for Systems [17]: the authors directed the design and ran every training/evaluation job, while the agent drafted and iterated on the implementation (code, scripts, and manuscript) under the authors' review. The full human-agent split is detailed in AI Workflow Disclosure.

II. Related Work

Our pipeline draws on a small set of well-established ingredients from the flow-matching, preference-learning, and music-generation literature. We close with a brief note on AI-driven research workflows, the methodological frame for our engineering loop.

a) Flow matching for audio generation: Flow matching [18] learns a continuous-time velocity field whose Euler integration maps prior noise to data. The Flux-style flow-matching transformer [9] provides our backbone, in the form of the FluxAudio-S baseline supplied by the challenge organizers [1]. *Classifier-free guidance* [11] combines a conditional and an unconditional pass at inference to amplify the conditioning signal.

b) Self-improvement via expert iteration: Expert iteration, in the ExIt [12] and ReST [13] formulations, alternates between sampling from the current policy and fine-tuning on top-quality samples. We use a one-round version filtered by our learned reward jointly with CLAP-text similarity.

c) Preference optimization: Direct preference optimization (DPO) [15] fits a policy directly to pairwise preferences without training a separate reward model. *CLAP-Ranked Preference Optimization* (CRPO), introduced in TangoFlux [14], adapts DPO to text-to-music by constructing preference pairs with a CLAP-text scorer, and we use the same procedure.

d) Music representations and preference data: Music-audio encoders include text-audio contrastive models (e.g., LAION-CLAP-Music [3]) and music-pretrained self-supervised models (e.g., MERT-v1-330M [8]). Open preference data for music has emerged across multiple sources, including Music Arena [19], MusicPrefs [20], AIME [21], and SongEval [22]. Our ranker pools all four with a RankNet [7] pairwise logistic loss (Section III-C). The backbone uses a pretrained BigVGAN vocoder [23] to decode 1D-Mel VAE latents, and vocal residuals are removed with Demucs [16].

e) AI-driven research workflows: The pattern of a human-defined objective with an LLM agent iterating against a programmatic evaluator has recently been formalized as AI-Driven Research for Systems [17], with closely related instances in FunSearch [24] (math) and AlphaEvolve [25] (algorithms). Our human-agent loop borrows the same high-level structure, with full disclosure in AI Workflow Disclosure.

III. Proposed Method

We describe the backbone, score-conditioning head, and preference ranker in this section, the training pipeline in Section III-D, and the inference procedure in Section III-E. The deployed system trains as v1 (Stages 1–2) and switches to v2 only at Stage 3 via cross-loading, justified in Section IV-B. Fig. 1 summarizes the pipeline.

TABLE I
Score-conditioning architecture comparison on the Jamendo-100 validation set (one generation per clip per input score). CLAP: CLAP-text cosine similarity. Score-r: Pearson correlation between input s and output reward. Δ_{OUT}: mean output reward at $s{=}{+}1.5$ minus that at $s{=}{-}0.5$ (higher = more steerable). Best per column in bold.

Variant	FAD-CLAP ↓	CLAP ↑	Score-r ↑	Δ_{out}
FluxAudio-S (baseline)	0.377	0.213	–	–
GlobalAdaLN (v1)	0.352	0.242	0.442	**0.942**
InputAdd (v2)	**0.337**	**0.249**	**0.524**	0.779
AudioPrepend (v3)	0.339	0.245	0.439	0.825
PerBlock AdaLN (v4)	0.347	0.243	0.446	0.856
TextPrepend (v5)	0.348	0.244	0.439	0.757

A. Backbone

The generative backbone is *FluxAudio-S*, the 120 M-parameter Flux-style flow-matching transformer [9] provided as the official challenge baseline [1]. It operates on 1D-Mel variational autoencoder (VAE) latents at 44.1 kHz (∼10 s per clip), with text conditioning via T5-Large [26] cross-attention and pooled LAION-CLAP [3] features through adaptive layer normalization (AdaLN). Audio is synthesized from latents by a pretrained BigVGAN vocoder [23]. We adopt the unconditional FluxAudio-S checkpoint provided by the challenge organizers and add only the score-conditioning head described in Section III-B, with no other layers modified.

B. Score-Conditioning Head

The reward scalar $s \in \mathbb{R}$ enters as a second conditioning input parallel to text. It is mapped to a 448-d embedding e_s via Fourier features [10] and an MLP with a zero-initialized final projection, so the generator at the start of training is identical to the unconditional backbone. We compared five injection strategies on the Jamendo-100 validation set (Table I). InputAdd (v2), which broadcast-adds e_s to every audio latent at the input projection ($z_i \leftarrow z_i + e_s$), leads on FAD-CLAP, CLAP score, and input-score correlation. The deployed model uses InputAdd (v2) at inference with weights warm-started from a GlobalAdaLN (v1) chain. v1 and v2 share an identical parameter graph and differ only in the forward roles of the score-related weights (Section IV-B). The score is null-dropped ($\varnothing_s{=}0$) with probability 0.1 during training, mirroring text CFG.

C. Pairwise Preference Ranker

Our preference ranker, TuneJury [6], is a twin pairwise model that maps any audio clip plus an optional text prompt to a single quality scalar. Each branch takes a 2048-d concatenation of LAION-CLAP-Music [3] audio (512), MERT-v1-330M [8] audio (1024), and LAION-CLAP-Music text (512). LAION-CLAP-Music supplies a caption-aligned semantic representation while MERT covers pitch, harmony, rhythm, and timbre that text-aligned encoders under-represent. The pairwise (rather than pointwise-regression) formulation matches the supervision: each of

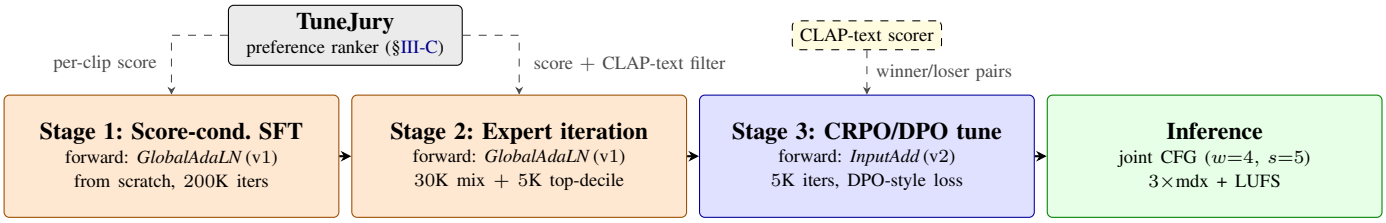

Fig. 1. **End-to-end system pipeline.** Box color marks the score-conditioning forward in use: orange for *GlobalAdaLN (v1)* (Stages 1 and 2), blue for *InputAdd (v2)* (Stage 3), and green for the deployed Inference endpoint (inherits v2 from Stage 3). GlobalAdaLN modulates the AdaLN parameters of every transformer block, and InputAdd broadcast-adds the reward embedding to every audio latent at the input projection only. Stages 1 and 2 train in the v1 forward because v1 converged more stably in pilots, and Stage 3 cross-loads the v1 weights into the v2 forward (parameter graphs are identical) and runs CRPO/DPO. The TuneJury score (gray dashed) is the training-time conditioning signal at Stage 1 and contributes to the top-decile filter at Stage 2, while the CRPO winner/loser pairs at Stage 3 are constructed from CLAP-text alignment (yellow dashed), following the standard CRPO procedure.

our four sources releases human votes as A-vs-B preferences, so the standard RankNet [7] pairwise logistic loss $\mathcal{L} = -\log\sigma(s(A)-s(B))$ consumes the labels directly. The score head is an MLP $2048\to1024\to512\to256\to128\to1$ with BatchNorm+ReLU+Dropout(0.5), trained on $\sim$22K pairs ($\sim$2K held out) pooled from Music Arena [19], MusicPrefs [20], AIME [21], and SongEval [22]. Held-out pairwise accuracy is 70.3%, with expected calibration error (ECE) 0.027. We use the ranker in two roles within the pipeline: (a) as a per-clip quality score that we attach to every training example and feed to the score-conditioning head, and (b) as the filter (jointly with a CLAP-text similarity score) that selects the top decile of self-generated samples for the expert-iteration fine-tune (Section III-D). The CRPO preference-tuning pass constructs its winner/loser pairs by CLAP-text alignment alone, following the standard CRPO procedure. Full design-space ablations are in the released repository [6].

D. Training Pipeline

The four training-time decisions of Section I are implemented as a three-stage chain: Stage 1 trains the score-conditioned backbone (operationalizing decisions (i) and (ii)), Stage 2 runs (iii) expert iteration, and Stage 3 runs (iv) CRPO. All three stages train on the same instrumental subset of MTG-Jamendo (vocals removed via Demucs), and only the data weighting and the loss change between stages.

a) Data: We start from the challenge-provided `jamendo_qwen.json` captions over the $\sim$55K-track MTG-Jamendo dataset [4], segment audio into 10 s clips ($\sim$535K clips), and apply Demucs [16] vocal separation to keep the instrumental stem only. Three reward columns per clip are computed with the ranker: `reward_score` (clip-level on full audio), `instrumental_reward_score` (clip-level on instrumental audio), and `track_reward_score` (track-level mean). The submitted model uses `instrumental_reward_score` as the conditioning signal because mixed-audio scoring partly tracks vocal presence. The generator learns to insert vocal-like artifacts to inflate the reward, hurting FAD-CLAP against an instrumental reference (FAD-CLAP 0.515 vs. 0.337 for two SFT runs trained with full-mix reward and instrumental-stem reward, respectively, under otherwise identical hyperparameters). Train-split statistics for the instrumental reward score are mean 0.62, std 0.59, p_5=−0.45, p_{95}=+1.46, max =+2.76. Validation and test splits hold out 200 tracks each. The original MTG-Jamendo tag vocabulary is not used directly: tag-derived genre/instrument/mood words appear inside the Qwen captions and reach the model only through the natural-language path.

b) Stage 1 (Score-Conditioned SFT): Stage 1 trains a score-conditioned backbone from scratch on the full 535K-clip set in the GlobalAdaLN (v1) forward, the most stable variant under our budget despite InputAdd (v2)'s slight edge in Table I. v1 carries through Stage 2 and is cross-loaded into v2 at Stage 3 (Section IV-B). Hyperparameters: AdamW, lr 10^{-4} (constant after a 1K-step warmup), effective batch 64, bf16, score-null dropout 0.1, 200K updates ($\sim$32 h on one NVIDIA RTX A5000), with EMA at $\sigma_{\text{rel}}\in\{0.05, 0.1\}$.

c) Stage 2 (Expert Iteration): Stage 2 fine-tunes the SFT checkpoint on a top-decile filter of its own outputs, in the spirit of expert iteration [12], [13]. We sample $\sim$600 clips from the SFT checkpoint at s=2.0, rank them by an equal-weight z-score blend of ranker reward and CLAP-text similarity, and keep the top 64 (reward mean +1.05, comparable to the upper $\sim$20% of the training distribution). The kept clips are then 5×-oversampled into the $\sim$535K-clip mixture and the checkpoint is fine-tuned for 30K steps at lr 10^{-5}, followed by a brief 5K-step polish at lr 10^{-6} on only the 64 top-decile clips.

d) Stage 3 (CRPO Preference-Tuning): Stage 3 switches to InputAdd (v2) and warm-starts backbone+`score_embed` from the v1 expert-iteration checkpoint via shape-matched partial loading that transfers all 203 keys (Section IV-B). It then runs CRPO [14] on 2,000 preference pairs: we score generated samples by CLAP-text alignment under each prompt and pair each high-CLAP sample with a low-CLAP sample under the same prompt. The DPO [15]-style loss is

$$\mathcal{L}_{\text{CRPO}} = -\log\sigma\Big(\beta\big(\Delta^{\pi}_{\text{win}} - \Delta^{\pi}_{\text{lose}}\big)\Big) + \lambda_{\text{FM}}\,\mathcal{L}^{\text{win}}_{\text{FM}} \qquad (1)$$

with $\Delta^{\pi}_{x} = \log[\pi(x)/\pi_{\text{ref}}(x)]$, β=2000 (the large β matches TangoFlux's CRPO scaling for flow-matching log-likelihood ratios, which are larger in magnitude than language-model token log-probs), λ_{FM}=1.0, lr 10^{-6}, 5K updates. The flow-matching auxiliary $\mathcal{L}^{\text{win}}_{\text{FM}}$ regularizes toward the warm-started reference. The resulting checkpoint is the submitted model.

e) Total compute: The full pipeline (SFT + expert-iteration + CRPO + ranker training) fits in approximately 40 GPU-hours on a single NVIDIA RTX A5000.

E. Inference Procedure

a) Joint classifier-free guidance: At inference, we apply classifier-free guidance [11] jointly on text and reward. Writing $\varnothing_t$ for the text-null and $\varnothing_s{=}0$ for the score-null, the velocity update is

$$\tilde{v} = v(x_t, t, \varnothing_t, \varnothing_s) + w\big[v(x_t, t, c, s) - v(x_t, t, \varnothing_t, \varnothing_s)\big], \quad (2)$$

where the same scalar w lifts text and score conditioning jointly relative to the doubly-unconditional baseline. We hold w fixed at 4.0 and the score scalar fixed at $s{=}5.0$, both selected on the SDD-100 validation set. We use a single fixed value at inference, not a sweep, consistent with the scope stated in Section I. The chosen $s{=}5.0$ lies above the training-time range of the reward score (max $+2.76$ on the train split, Section III-D). A full analytical study of the score-response curve under extrapolation is left to future work. Sampling uses 25 Euler steps (linear schedule $\sigma_i = 1 - i/25$), a seed fixed per submission, and the prompt prefix "`high quality instrumental music, `". End-to-end wall-clock is 0.5–0.8 s per 10 s clip on a single NVIDIA RTX A5000.

b) Source separation and loudness normalization: Two lightweight post-processing steps consistently improve metrics on internal validation. First, we pass each generated wav through three sequential applications of Demucs's `mdx_extra` model and keep the residual "no-vocals" track. Even with the "`instrumental music`" prefix, the score-conditioned generator occasionally produces vocal-like residuals that pollute FAD-CLAP against an instrumental reference, and the three-pass separator removes them. Second, we loudness-normalize the result to -16.5 LUFS via the ITU-R BS.1770 [27] algorithm with a true-peak ceiling at -1 dB. The LUFS target was selected on the validation set to minimize FAD-CLAP averaged across prompts, and values in the range -15 to -18 LUFS perform similarly.

c) Submitted configurations: We submit two configurations, Sub. 1 (seed 42) and Sub. 2 (seed 55), which share backbone weights and the post-processing pipeline above and differ only in the random seed used at inference.

IV. Experiments

We report internal validation on SDD-100, evaluated against the SDD-706 reference (both introduced in Section I). FAD-CLAP and CLAP score both use the LAION-CLAP-Music checkpoint `music_audioset_epoch_15_esc_90.14.pt` on 10-second clips, matching the official objective-metric protocol. FAD-CLAP is a distribution-level statistic (one value per condition); CLAP score and Reward are per-prompt and support paired-t tests. Throughout this section, *Reward* denotes the mean output of our preference ranker (Section III-C), and unless noted otherwise all rows use the same single-value inference protocol ($s{=}5.0$, $w{=}4.0$, 25 Euler steps, prefix prompt, seed 42 unless stated, 3×mdx_extra, -16.5 LUFS). This SDD-706 protocol is distinct from the architecture-selection protocol of Table I (Jamendo-100 reference, no post-processing), and absolute values across the two are not directly comparable.[1]

TABLE II
CUMULATIVE ABLATION ALONG THE DEPLOYED CHAIN ($N{=}100$ SDD PROMPTS). SUB. 2 IS THE SEED-55 SIBLING OF SUB. 1 (SAME CHAIN). †: PAIRED-t IMPROVEMENT OVER THE PREVIOUS ROW (ONE-SIDED, $p{<}0.05$). BEST PER COLUMN IN BOLD.

#	Pipeline (cumulative)	FAD-CLAP ↓	CLAP ↑	Reward ↑
0	FluxAudio-S (baseline)	0.5998	0.230	−0.392
1	Score-conditioned SFT (v1)	0.4681	0.262	+0.028
2	+ Expert iteration	0.4319	0.290†	+0.524†
3	+ Cross-load to v2 forward	0.4272	0.283	+0.535
4	+ CRPO (= Sub. 1, seed 42)	**0.4238**	0.285	+0.533
	Sub. 2 (seed 55)	0.4370	**0.300**	**+0.550**

A. Cumulative Stage Ablation

We anchor the chain at the unconditional FluxAudio-S checkpoint provided by the challenge organizers (Row 0, generated without score conditioning) and add each pipeline step in order, measuring the marginal contribution against the previous row (Table II). Only step 2 (expert iteration) reaches paired-t significance on either CLAP score or Reward, while steps 3 and 4 each leave the per-prompt distribution within paired-t noise of the previous row, in agreement with the cross-mechanism findings.

B. Cross-Mechanism Ablation

We treat the score-conditioning mechanism as a separable knob from the trained weights and run an 8-cell factorial: {*SFT-only*, *Chain-end*} × {v1 weights, v2 weights} × {v1 forward, v2 forward}, plus the two submitted configurations (Table III). *SFT-only* is the post-Stage-1 checkpoint, *Chain-end* is post-Stage-2 (before CRPO), and *Hybrid (submitted)* is post-Stage-3 (CRPO over Chain-end v1 → v2). Cross-loading uses `state_dict.load(strict=False)`, and v1 and v2 share an identical 203-key parameter graph.

C. Inference-Time Score Sensitivity

To check whether the inference-time score scalar does visible work on the deployed Hybrid (Sub. 1) checkpoint, we sweep $s \in [0, 6]$ on the SFT-only and Hybrid checkpoints (Fig. 2).

The two curves diverge sharply. On the SFT-only checkpoint the score scalar moves Reward from $+0.16$ (at $s{=}0$) to $+0.47$ (at $s{=}2$) with Spearman $\rho{=}1.0$ across the training range $s \in [0, 2]$, confirming that score conditioning at training time produced a backbone whose output reward tracks the input scalar monotonically. On the submitted Hybrid checkpoint, however, s is nearly inert: Reward already sits at $+0.54$ at $s{=}0$, the entire $s \in [0, 6]$ range varies Reward by less than

[1] Under the challenge's hidden Jamendo reference set, our submission (`e02`) scored FAD 0.498, CLAP 0.270, CCS 0.763 [1].

TABLE III
CROSS-MECHANISM ABLATION (N=100 SDD PROMPTS). * MARKS CELLS STATISTICALLY TIED WITH SUB. 1 ON THE PER-PROMPT MARGIN (PAIRED-t, $p \geq 0.05$); UNMARKED CELLS ARE SIGNIFICANTLY WORSE THAN SUB. 1 ON THAT METRIC. BEST PER COLUMN IN BOLD. HYBRID ROWS ARE SUB. 1 (SEED 42) AND ITS SEED-55 SIBLING SUB. 2.

Stage	Weights → Forward	FAD-CLAP ↓	CLAP ↑	Reward ↑
SFT only	v1 → v1 (native)	0.4681	0.262	+0.028
	v1 → v2 (cross)	0.4456	0.265	+0.009
	v2 → v1 (cross)	0.6846	0.202	−0.500
	v2 → v2 (native)	0.4442	0.266	+0.282
Chain end	v1 → v1 (native)	0.4319	0.290*	+0.524*
	v1 → v2 (cross)	0.4272	0.283*	+0.535*
	v2 → v1 (cross)	0.6952	0.198	−0.518
	v2 → v2 (CRPO)	0.4695	0.265	+0.244
Hybrid (submitted)	v1 → v2 (Sub. 1, seed 42)	**0.4238**	0.285	+0.533
	v1 → v2 (Sub. 2, seed 55)	0.4370	**0.300**	**+0.550**

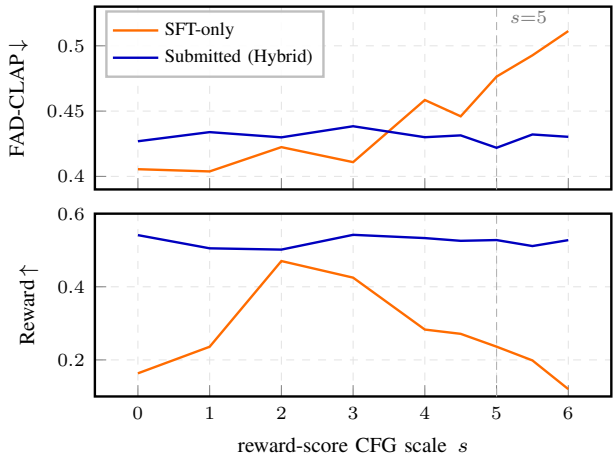

Fig. 2. **Inference-time score sweep on** 100 **SDD prompts. SFT-only** (orange) tracks the reward monotonically within its training range: Spearman ρ=1.0 on Reward across $s \in [0, 2]$, with Reward rising from +0.16 to +0.47, and past s=3 the curve bends and FAD-CLAP rises. **Submitted** (blue, Hybrid Sub. 1 from Table III) is essentially flat in both metrics across the full $s \in [0, 6]$ range (Reward range 0.04, Pearson $r \approx 0$), i.e. the inference knob has saturated. Dotted line marks the deployed value s=5.

0.05 and FAD-CLAP by less than 0.02 (Pearson $r \approx 0$ between s and Reward), and the FAD-CLAP optimum at s=5 (0.4219) is within seed-noise of the unconditioned s=0 pass (0.4269). We hold s=5.0 in our submission because validation selected it, not because the inference-time score is the lever moving the model. The lever has been absorbed into the weights upstream by expert iteration and CRPO, which leaves the inference scalar with little remaining headroom.

D. Engineering Observations

We provide three takeaways from our development process.

a) Score conditioning matters at training time and is saturated at inference: Score-conditioned variants improve FAD-CLAP by 0.025–0.040 absolute over the unconditional baseline at the SFT stage (Table I), but the deployed Hybrid checkpoint shows a flat s response across $s \in [0, 6]$ (Fig. 2, blue). The score signal therefore does its real work *during* training: the backbone absorbs reward into its weights, and expert iteration and CRPO take up what little inference-time steerable margin remained. The chain in effect trades the SFT-only model's working s knob for a higher absolute baseline, with Sub. 1 at any s matching or exceeding the SFT-only peak.

b) Mechanism transfer is asymmetric: v1 → v2 is benign, v2 → v1 collapses: Table III shows a sharp asymmetry. Loading a v1-trained checkpoint into the v2 (InputAdd) forward stays within 0.02 FAD-CLAP of the v1-native cell at both stages (Chain-end v1→v2 Reward +0.535 vs. v1-native +0.524). The reverse cross collapses to FAD-CLAP ∼0.69 and Reward ∼−0.50. InputAdd is additive on audio tokens, so unfamiliar score weights dampen rather than distort. GlobalAdaLN modulates every layer, and a v2-trained `score_embed` feeds the AdaLN with patterns far outside the training distribution. This justifies our hybrid direction of warm-starting a v2-architecture CRPO from a v1 backbone.

c) Expert iteration is the dominant chain contributor, while CRPO adds noise-level gain at this scale: The biggest delta in Table III comes from expert iteration on the v1 chain (SFT-only → Chain-end: FAD-CLAP −0.0362, CLAP +0.028, Reward +0.496). The v2 chain regresses on the same axis (0.4442 → 0.4695, Reward +0.282 → +0.244), as v2 expert iteration plus CRPO did not converge cleanly under our budget. Adding 5K CRPO steps on top of the v1 chain (Chain-end v1 → v2 → Sub. 1) shifts FAD-CLAP by −0.003, CLAP by +0.002, and Reward by −0.002. Neither per-prompt difference reaches paired-t significance at p<0.05.

V. CONCLUSION

We submitted a 40 GPU-hour entry to the ICME 2026 ATTM Grand Challenge efficiency track on the 120 M-parameter FluxAudio-S baseline, with TuneJury supplying a learned human-preference reward used both as a training-time conditioning signal and as a sample-selection criterion. Three findings emerge from the per-stage ablation. (a) Training-time reward conditioning is a functional steering axis (FAD-CLAP 0.025–0.040 at SFT), but its effect is absorbed into the weights by chain-end and the inference-time scalar saturates. (b) Mechanism transfer is asymmetric: v1 GlobalAdaLN → v2 InputAdd cross-loads benignly (and we deploy this hybrid), while the reverse collapses. (c) Reward-filtered expert iteration is the dominant chain contributor (−0.0362 FAD-CLAP on the v1 chain), with the CRPO pass at noise-level gain. A full analytical study of the score-response curve under extrapolation and a cross-family replication on other backbones are left to future work.

AI WORKFLOW DISCLOSURE

Building on the Workflow note in Section I, we record the human-agent split for transparency.

a) Tool and task specification: Claude Code CLI with Anthropic's *Claude Opus* 4.6 [28] and 4.7 [29]. Tasks were issued as small, conversational natural-language requests rather than a formal specification, and the agent had no autonomous evaluation budget against a fixed objective.

b) Direction (human, with agent assistance): The architectural, training-data, evaluation, and post-processing choices reported in this paper (the five score-conditioning variants, the v1 $\rightarrow$ v2 hybrid, top-decile expert iteration, the CRPO pass, the instrumental-stem reward, the $3\times$mdx_extra source separation, the -16.5 LUFS target, and the single-scale joint CFG) were proposed by the human authors, who also ran every training, generation, and evaluation job and validated the results. The agent's conceptual contribution was mainly mapping author-described procedures onto existing literature (e.g., recognizing the top-decile filter as expert iteration [12], [13] and the preference-pair pass as CRPO [14]) and suggesting baseline hyperparameters and small variants during review.

c) Implementation (agent): The agent wrote most of the line-level code: score-conditioning heads, expert-iteration sampling/filtering, CRPO loop, post-processing scripts, evaluation harnesses, and the TikZ/pgfplots source for Fig. 1 and Fig. 2. It also drafted and edited the manuscript, maintained the bibliography, and tuned LaTeX layout. Figure layouts and color choices were converged on iteratively rather than in a single pass.

d) Supervision: The authors reviewed every commit, accepted or revised every edit in the manuscript, and gated all Overleaf master pushes. The agent did not run training or evaluation jobs autonomously.

REFERENCES

[1] 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 *International Conference on Multimedia and Expo, Grand Challenge Paper*, 2026.

[2] Kevin Kilgour, Mauricio Zuluaga, Dominik Roblek, and Matthew Sharifi, "Fréchet audio distance: A reference-free metric for evaluating music enhancement algorithms," in *Proceedings of Interspeech*, 2019.

[3] Yusong Wu, Ke Chen, Tianyu Zhang, Yuchen Hui, Marianna Nezhurina, Taylor Berg-Kirkpatrick, and Shlomo Dubnov, "Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation," in *Proceedings of ICASSP*, 2023.

[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, ICML*, 2019.

[5] Ilaria Manco, Benno Weck, Seungheon Doh, Minz Won, Yixiao Zhang, Dmitry Bogdanov, Yusong Wu, Ke Chen, Philip Tovstogan, Emmanouil Benetos, Elio Quinton, György Fazekas, and Juhan Nam, "The song describer dataset: a corpus of audio captions for music-and-language evaluation," in *Machine Learning for Audio Workshop, NeurIPS*, 2023.

[6] "TuneJury: An open metric for improving music generation preference alignment," https://github.com/yonghyunk1m/tune-jury, 2026, Source code repository.

[7] Christopher Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender, "Learning to rank using gradient descent," in *Proceedings of ICML*, 2005.

[8] Yizhi Li, Ruibin Yuan, Ge Zhang, Yinghao Ma, Xingran Chen, Hanzhi Yin, Chenghao Xiao, Chenghua Lin, Anton Ragni, Emmanouil Benetos, Norbert Gyenge, Roger Dannenberg, Ruibo Liu, Wenhu Chen, Gus Xia, Yemin Shi, Wenhao Huang, Zili Wang, Yike Guo, and Jie Fu, "MERT: Acoustic music understanding model with large-scale self-supervised training," in *Proceedings of ICLR*, 2024.

[9] Zhengcong Fei, Mingyuan Fan, Changqian Yu, and Junshi Huang, "FLUX that plays music," *arXiv preprint arXiv:2409.00587*, 2024.

[10] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng, "Fourier features let networks learn high frequency functions in low dimensional domains," in *Proceedings of NeurIPS*, 2020.

[11] Jonathan Ho and Tim Salimans, "Classifier-free diffusion guidance," *arXiv preprint arXiv:2207.12598*, 2022.

[12] Thomas Anthony, Zheng Tian, and David Barber, "Thinking fast and slow with deep learning and tree search," in *Proceedings of NeurIPS*, 2017.

[13] Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, Wolfgang Macherey, Arnaud Doucet, Orhan Firat, and Nando de Freitas, "Reinforced self-training (ReST) for language modeling," *arXiv preprint arXiv:2308.08998*, 2023.

[14] Chia-Yu Hung, Navonil Majumder, Zhifeng Kong, Ambuj Mehrish, Amir Ali Bagherzadeh, Chuan Li, Rafael Valle, Bryan Catanzaro, and Soujanya Poria, "TangoFlux: Super fast and faithful text to audio generation with flow matching and Clap-ranked preference optimization," in *Proceedings of ICLR*, 2026.

[15] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn, "Direct preference optimization: Your language model is secretly a reward model," in *Proceedings of NeurIPS*, 2023.

[16] Alexandre Défossez, Nicolas Usunier, Léon Bottou, and Francis Bach, "Music source separation in the waveform domain," *arXiv preprint arXiv:1911.13254*, 2019.

[17] Audrey Cheng, Shu Liu, Melissa Pan, Zhifei Li, Bowen Wang, Alex Krentsel, Tian Xia, Mert Cemri, Jongseok Park, Shuo Yang, Jeff Chen, Lakshya Agrawal, Aditya Desai, Jiarong Xing, Koushik Sen, Matei Zaharia, and Ion Stoica, "Barbarians at the gate: How AI is upending systems research," *arXiv preprint arXiv:2510.06189*, 2025.

[18] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le, "Flow matching for generative modeling," in *Proceedings of ICLR*, 2023.

[19] Yonghyun Kim, Wayne Chi, Anastasios Angelopoulos, Wei-Lin Chiang, Koichi Saito, Shinji Watanabe, Yuki Mitsufuji, and Chris Donahue, "Music arena: Live evaluation for text-to-music," in *NeurIPS Creative AI Track*, 2025.

[20] Yichen Huang, Zachary Novack, Koichi Saito, Jiatong Shi, Shinji Watanabe, Yuki Mitsufuji, John Thickstun, and Chris Donahue, "Aligning text-to-music evaluation with human preferences," in *Proceedings of ISMIR*, 2025.

[21] Florian Grötschla, Ahmet Solak, Luca A. Lanzendörfer, and Roger Wattenhofer, "Benchmarking music generation models and metrics via human preference studies," in *Proceedings of ICASSP*, 2025.

[22] Jixun Yao, Guobin Ma, Huixin Xue, Huakang Chen, Chunbo Hao, Yuepeng Jiang, Haohe Liu, Ruibin Yuan, Jin Xu, Wei Xue, Hao Liu, and Lei Xie, "SongEval: A benchmark dataset for song aesthetics evaluation," *arXiv preprint arXiv:2505.10793*, 2025.

[23] Sang-gil Lee, Wei Ping, Boris Ginsburg, Bryan Catanzaro, and Sungroh Yoon, "BigVGAN: A universal neural vocoder with large-scale training," in *Proceedings of ICLR*, 2023.

[24] Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli, and Alhussein Fawzi, "Mathematical discoveries from program search with large language models," *Nature*, vol. 625, pp. 468–475, 2024.

[25] Alexander Novikov, Ngân Vũ, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco J. R. Ruiz, Abbas Mehrabian, M. Pawan Kumar, Abigail See, Swarat Chaudhuri, George Holland, Alex Davies, Sebastian Nowozin, Pushmeet Kohli, and Matej Balog, "AlphaEvolve: A coding agent for scientific and algorithmic discovery," *arXiv preprint arXiv:2506.13131*, 2025.

[26] 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," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020.

[27] International Telecommunication Union, "Recommendation ITU-R BS.1770-4: Algorithms to measure audio programme loudness and true-peak audio level," ITU-R Recommendation, 2015.

[28] Anthropic, "Claude Opus 4.6 system card," https://anthropic.com/claude-opus-4-6-system-card, 2026.

[29] Anthropic, "Claude Opus 4.7 system card," https://anthropic.com/claude-opus-4-7-system-card, 2026.