REINFORCEMENT LEARNING FOR MUSIC GENERATION WITH PAIRWISE REWARD MODELING

Anonymous authors
Paper under double-blind review

ABSTRACT

Improving text-to-music models through reinforcement learning (RL) requires a reward model that can clearly quantify the quality of a text-music pair. In this domain, this usually includes (i) how high-quality the outputs sound, and (ii) how well they align with the provided text. We introduce *DuetJudge*, an on-policy pairwise reward model that learns to directly compare two instrumental pieces conditioned on the same caption, without assigning independent scalar scores to each. *DuetJudge* is functionally a ternary classifier that outputs either one piece wins or they are a tie. Particularly, *DuetJudge* prioritizes text-music alignment, and then considers perceptual quality of the music. *DuetJudge* achieves 71% agreement with human preferences on our internal test set, outperforming all existing score-based baselines. To use *DuetJudge* as a reward model in text-to-music RL, we (i) design a reward score based on aggregated pairwise preference probabilities, and (ii) introduce classifier-free guidance (CFG)-aware policy optimization for text-to-music models trained with CFG. We test *DuetJudge* using multiple in-distrubtion and out-of-distribution text-to-music models, and find systematic improvements in both automatic metrics and human ratings. Ablation studies further favor direct pairwise rewards over a Bradley–Terry scalar baseline under the same training settings. These results confirm the effectiveness of our proposed *DuetJudge*, our data curation and augmentation methods, our reward model training recipes. and our text-to-music RL training recipes. An anonymous demo is available at https://anonymous.4open.science/w/duetjudge-demo-8E08/.

1 INTRODUCTION

What makes music good? A compelling melody, an expressive performance, and clear production can all contribute, yet their importance depends on the music and the listener. Music-listening studies associate liking with situational appropriateness (North & Hargreaves, 1996) and the attainment of listening goals (Schäfer, 2016). For text-to-music generation, we focus on a concrete requirement: the music should fulfill the user's request. A polished, upbeat pop arrangement is not a successful response to a request for "a slow, mournful solo cello piece" (Figure 1). Conversely, matching the requested instrument and mood does not make audible artifacts irrelevant. A useful judge must therefore consider both whether the music fulfills the request and whether it sounds good.

Existing pointwise music evaluators provide useful signals for these judgments. MuQ-MuLan (Zhu et al., 2025) measures music–text similarity, while Audiobox Aesthetics (Tjandra et al., 2025) predicts audio-only aesthetic scores. CMI-RM (Ma et al., 2026) estimates musicality and alignment separately, and TuneJury (Kim et al., 2026) predicts an overall preference score. Although the latter two use *pairwise supervision*, each scores a candidate without observing its alternative; pairwise training labels do not imply *joint comparison* at inference. A direct judge can instead inspect both clips to determine which audible differences matter under the caption. We investigate whether learning the comparison itself can provide better music judgments and more effective rewards for generation.

However, jointly comparing two clips is not sufficient unless the preference criterion is explicit. Research on preference construction shows that expressed preferences can depend on the elicitation procedure (Slovic, 1995). In music generation, overall listening appeal need not coincide with request fulfillment. MusicRL (Cideron et al., 2024) reports that removing text tokens during preference reward-model training leaves prediction accuracy largely unchanged, suggesting that text adherence

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

was not a primary driver of preference in its user-feedback setting. To prioritize *request fulfillment*, we adopt a task-specific *alignment-first, quality-second* criterion: a meaningful advantage in caption alignment takes precedence, while perceptual quality distinguishes comparably aligned candidates. This makes the intended priority explicit in the supervision rather than assuming it will emerge from unconstrained overall preferences.

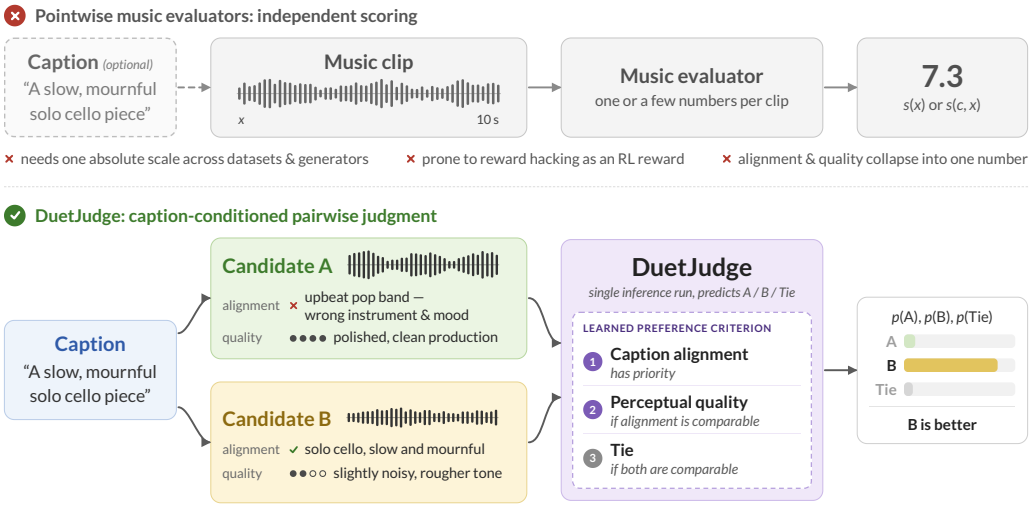


Figure 1: **From independent scoring to direct pairwise judgment.** Top: pointwise reward models evaluate each music clip independently. Bottom: *DuetJudge* jointly compares two candidates conditioned on the same caption, prioritizing caption alignment and then perceptual quality, with an explicit TIE option. Candidate B is preferred because it fulfills the requested instrument, tempo, and mood despite its less polished production.

We introduce ***DuetJudge***, a caption-conditioned pairwise reward model for instrumental music built on Audex-2B (Kong et al., 2026). Given a caption and two clips, it jointly evaluates the candidates and predicts *A*, *B*, or TIE, allowing a tie when neither candidate is meaningfully preferable or the audible evidence is insufficient.

To teach which differences matter, we construct three complementary supervision families. *Holistic Preference* comparisons cover real and generated candidates whose musical content and perceptual quality may differ simultaneously. *Fine-grained Alignment* comparisons target individual captioned attributes, such as instrument or tempo. *Perceptual Quality* comparisons vary the degradation applied to the same source clip, isolating quality differences while retaining the underlying musical content. After an initial music-understanding stage, we specialize the model using decision-only supervision, balancing these families to retain sensitivity to caption constraints without letting easily detected production defects dominate. We include explicit tie supervision and both candidate orders.

Evaluated as a standalone judge, *DuetJudge* achieves 71% agreement with human preferences, outperforming all evaluated baselines. Beyond standalone evaluation, we assess its utility as a frozen reward model for reinforcement learning (RL). We use *DuetJudge* to optimize Audex-2B and Audex-30B-A3B in-domain, and evaluate out-of-domain reward transfer on ETTA (Lee et al., 2025), a diffusion-based generator whose outputs are excluded from *DuetJudge* training. Across all three generators, optimization with the same frozen judge improves over the corresponding initialization on both automatic and human ratings. For Audex-2B, direct pairwise rewards also outperform a Bradley–Terry scalar reward-model baseline when both are optimized under the same optimization setup.

Our contributions are fourfold:

- **Direct pairwise music reward modeling.** We introduce *DuetJudge*, a caption-conditioned judge for instrumental music that jointly compares two candidates rather than scoring them independently. It predicts a preference for *A* or *B*, or a TIE under an explicit *alignment-first, quality-*

second criterion. As a standalone judge, it achieves 71% agreement with human preferences, outperforming all evaluated baselines.

- **Large-scale judgment data.** We construct 287K comparisons spanning holistic preferences, fine-grained alignment, and controlled quality degradations, with explicit tie supervision.
- **Effective rewards across scales and generators.** Using *DuetJudge* as a frozen RL judge improves Audex-2B, Audex-30B-A3B, and diffusion-based ETTA over their initializations on all automatic metrics and both human ratings. For Audex, we develop classifier-free guidance (CFG)-aware policy optimization, coupling direct guided-policy updates with unconditional-branch regularization. ETTA's gains demonstrate reward transfer to an out-of-domain generator.
- **Practical recipes for music reward modeling and RL.** Our ablations show that how music preferences are represented matters for downstream optimization: direct comparisons outperform independent scalar scoring, while retaining ties improves musicality and caption alignment. Comparisons among a policy's own same-caption generations also provide effective rewards without real-audio references.

2 JUDGMENT DATA

We construct judgment data under the *request-fulfillment* criterion introduced above. Each example is a caption-conditioned comparison

$$(c, x_A, x_B, y), \qquad y \in \{\mathrm{A}, \mathrm{B}, \textsc{Tie}\},$$

where c is a caption, x_A and x_B are 10-second music clips, and y identifies the preferred candidate in the presented order or a tie. We follow an *alignment-first, quality-second* rule: a meaningful advantage in caption alignment determines the winner, while perceptual quality distinguishes comparably aligned candidates. We assign TIE when the candidates are comparable in both alignment and quality, or when the audible evidence is insufficient. Figure 2 illustrates the three supervision families, whose construction counts appear in Table 1.

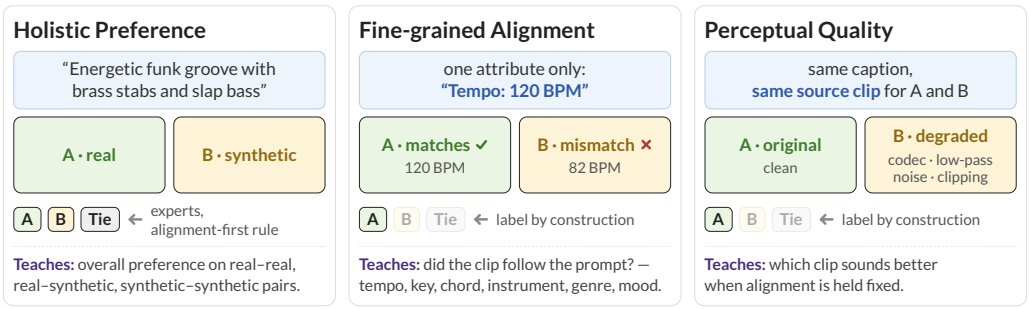


Figure 2: Three complementary forms of judgment supervision. Holistic pairs teach the full preference hierarchy; fine-grained pairs target captioned attributes; quality pairs compare degradations of the same source clip.

2.1 HOLISTIC PREFERENCE

Holistic comparisons teach the full preference rule when musical content and perceptual quality vary. We first use non-vocal samples from academic sources and Audex models (Kong et al., 2026) to construct *real–real*, *real–synthetic*, and *synthetic–synthetic* pairs, including comparisons across generators and between independent samples from the same generator. We then use MuQ-MuLan (Zhu et al., 2025) and Audiobox-PQ (Tjandra et al., 2025) for preliminary candidate screening, followed by expert labeling of the preference decisions. Candidate positions are randomized, and reversed comparisons exchange A/B labels while preserving TIE. We refer to Appendix A.1 for more details.

2.2 FINE-GRAINED ALIGNMENT

Fine-grained comparisons target individual caption constraints using annotated real recordings and controlled prompt variations. For real recordings, each caption specifies one of *tempo, instrument,*

key, chord progression, genre, or mood. Verified reference labels identify non-ties where only one candidate matches the captioned attribute; ties, where included, require both candidates to match and have similar Audiobox-PQ scores. Auxiliary models verify local consistency without replacing the reference annotations, while attribute-specific checks exclude ambiguous negatives. The recordings need not otherwise match: the construction isolates the captioned criterion, not every audible property. Together, these fine-grained examples preserve sensitivity to explicit caption constraints while prioritizing alignment with the caption over perceptual quality. Appendix A.2 describes the attribute-specific verification and pair-construction procedures.

2.3 Perceptual Quality

These comparisons target audible artifacts by degrading one or both variants of the same real or generated source clip, retaining its caption and underlying music. *Normal-versus-degraded* pairs prefer the candidate without added degradation; *light-versus-heavy* pairs prefer the lighter variant within one degradation family. All comparisons are non-ties and we do not assume an ordering of severity across degradation families. For generated sources, both candidates inherit the same output's characteristic texture, so the task targets added degradation rather than real-versus-generated discrimination. Appendix A.3 details source screening, shared preprocessing, degradation presets, and severity sampling.

Table 1: Construction scale before order augmentation and training-time repetition.

Data family	**Label source**	**# Samples**	**Audio hours**	**Audio tokens**
Holistic Preference	Expert labels	163,327	908	82 M
Fine-grained Alignment	Controlled comparisons	65,152	362	33 M
Perceptual Quality	Same-source degradation ordering	59,239	329	30 M
Total		287,718	1,599	145 M

3 Reward Modeling: *DuetJudge*

Given a caption c and an ordered pair of music clips (x_A, x_B), *DuetJudge* jointly evaluates the two candidates under the alignment-first, quality-second criterion and generates a decision $y \in \{\mathrm{A}, \mathrm{B}, \textsc{Tie}\}$. Let ϕ denote the judge parameters and $y_{1:T_y}$ the token sequence representing the decision. We optimize a decision-only autoregressive objective:

$$\mathcal{L}_{\text{judge}}(\phi) = -\sum_{t=1}^{T_y} \log p_\phi(y_t \mid c, x_A, x_B, y_{<t}). \tag{1}$$

Only tokens corresponding to the final decision contribute to the loss. Unlike pointwise reward models that assign an independent scalar score to each clip, *DuetJudge* directly learns a caption-conditioned comparison rule over the two candidates.

3.1 Two-Stage Training

We train *DuetJudge* in two stages, starting from Audex-2B (Kong et al., 2026). Stage I strengthens music understanding through supervised fine-tuning on music, audio, and text instruction data. Stage II specializes this checkpoint on the three comparison families in Section 2, using decision-only supervision to learn the alignment-first, quality-second preference rule.

The Stage-II mixture emphasizes same-caption synthetic and fine-grained comparisons rather than sampling in proportion to dataset size, while downweighting quality-only supervision so that production artifacts do not dominate subtler alignment differences. We retain explicit TIE supervision, include both candidate orders with label-consistent reversal, and interleave non-comparison understanding and instruction data to retain underlying capabilities. Appendices B.3 and D.1 provide the sampling rationale, data composition, and training configuration.

3.2 EVALUATING *DuetJudge* AS A MUSIC JUDGE

To assess whether *DuetJudge* has learned the intended alignment-first, quality-second criterion, we evaluate it as a standalone music judge on three complementary tests: human-labeled overall preference, alignment–quality conflicts, and same-source quality degradations.

Music evaluators. We compare against six pointwise baselines: Audiobox Aesthetics (Tjandra et al., 2025) (CE and PQ) for audio-only aesthetics; CMI-RM (Ma et al., 2026) (Musicality and Alignment) for musicality and caption alignment; MuQ-MuLan (Zhu et al., 2025) for music–text similarity; and TuneJury (Kim et al., 2026) for overall preference. Appendix F.1 provides further details on these evaluators and their scoring dimensions.

Pairwise decisions. Each baseline independently assigns scalar scores s_A and s_B to the two candidates, using the caption when supported. We predict TIE if $|s_A - s_B| < \delta$, with the threshold δ optimized separately for each baseline; otherwise, the higher-scoring candidate wins. For *DuetJudge*, we use decision probabilities rather than decoded labels, averaging both candidate orders after aligning the *A*/*B* labels. We predict TIE if $|P(\mathrm{A}) - P(\mathrm{B})| < 0.05$ or $P(\mathrm{TIE}) > 0.25$; otherwise, we choose *A* or *B* according to which has the higher probability.

Overall preference. We evaluate agreement with human preferences on same-caption pairs of Audex-2B outputs. We sample a subset of captions from the Song Describer Dataset (Manco et al., 2023) and, for each sampled caption, generate two music candidates using different random seeds. Human listeners compare these generated pairs and assign a preference label in $\{\mathrm{A}, \mathrm{B}, \mathrm{TIE}\}$. This same-caption, same-generator setting tests preference discrimination among stochastic outputs from the generator used to initialize subsequent Audex-2B policy optimization. We report exact-label agreement overall and separately for ties and non-ties, with annotation details provided in Appendix F.2. As shown in Table 2, *DuetJudge* achieves 71.0% overall agreement, outperforming all evaluated baselines both overall and on the tie and non-tie subsets.

Table 2: Music-judge evaluation on overall preference, alignment–quality conflicts, and same-source quality degradations. Overall preference reports agreement with human labels; Conflict and Quality report accuracy against construction-based reference labels. All scores are percentages (↑).

Judge	Overall preference (agreement)			Conflict	Quality
	Overall	Tie	Non-tie	Acc.	Acc.
Audiobox-CE	42.0	25.0	46.3	18.8	82.5
Audiobox-PQ	46.0	25.0	51.2	12.5	**95.0**
CMI-RM Musicality	42.0	20.0	47.5	26.3	68.8
CMI-RM Alignment	41.0	25.0	45.0	28.7	65.0
MuQ-MuLan	54.0	50.0	55.0	46.3	42.5
TuneJury	53.0	30.0	58.8	52.5	47.5
***DuetJudge* (ours)**	**71.0**	**60.0**	**73.8**	**66.3**	86.3

Alignment–quality conflicts. We construct conflict pairs only from non-tied overall-preference examples whose preferred candidate was selected at the caption-alignment stage of human annotation, excluding cases decided by the quality tiebreaker. We degrade the perceptual quality of this better-aligned candidate and compare it with the unmodified alternative, retaining the degraded candidate as the construction-based reference choice. Table 2 shows that *DuetJudge* achieves the highest accuracy of 66.3%, exceeding the strongest baseline, TuneJury (52.5%), by 13.8 percentage points. In contrast, Audiobox-CE and Audiobox-PQ achieve only 18.8% and 12.5%, respectively. Their audio-only scores reflect listening quality, but not how well the music matches the caption.

Perceptual quality. We construct same-source quality pairs by comparing the same degraded winners with their original versions, using the originals as the reference choices. *DuetJudge* achieves 86.3% accuracy, second to Audiobox-PQ (95.0%) and ahead of Audiobox-CE (82.5%). MuQ-MuLan and TuneJury obtain only 42.5% and 47.5%, respectively, showing little ability to distinguish the originals from their degraded versions. Together, these results suggest that *DuetJudge*'s stronger

performance on alignment–quality conflicts is not simply due to ignoring quality: it recognizes perceptual defects while giving caption alignment priority.

4 POLICY OPTIMIZATION

Having trained *DuetJudge* to make alignment-first, quality-second comparisons, we next test whether optimizing against its judgments improves text-to-music generation. We keep the judge fixed and aggregate its pairwise preference probabilities among candidates for the same caption into relative rewards. We first optimize autoregressive Audex with a CFG-aware RL algorithm that uses the guided policy consistently for rollout generation and likelihood evaluation, while separately regularizing the unconditional branch. We then apply the same frozen judge to diffusion-based ETTA, whose outputs are absent from *DuetJudge*'s training data, to further assess reward transfer to an unseen generator with a different architecture. Below, we introduce the shared audio-level reward construction before describing the model-specific optimization procedures.

Pairwise reward construction. For each caption c, we sample G candidate audio clips $x_1, \ldots, x_G$ and compare every pair in both presentation orders using the frozen *DuetJudge*. For notational simplicity, we omit the caption index c when referring to candidates within the same group, writing x_i and r_i in place of $x_{c,i}$ and $r_{c,i}$.

Here, i and j identify candidates within the group, whereas A and B denote their presentation slots in a particular comparison. For the ordered pair (x_i, x_j), the judge input is instantiated as $(x_A, x_B) = (x_i, x_j)$. Let $q_\phi(y \mid c, x_i, x_j)$ denote the judge's decision-probability estimates, normalized over $y \in \{\mathrm{A}, \mathrm{B}, \textsc{Tie}\}$. Appendix C.1 describes probability extraction. We average the probability of preferring candidate i over candidate j across the two presentation orders:

$$p_{ij} = \frac{1}{2}\left[q_\phi(\mathrm{A} \mid c, x_i, x_j) + q_\phi(\mathrm{B} \mid c, x_j, x_i)\right].$$

The first term presents x_i as A, while the second presents it as B; both therefore express a preference for the same candidate x_i. Each candidate receives its average preference margin against others:

$$r_i = \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} (p_{ij} - p_{ji}), \tag{2}$$

where $\mathcal{N}_i$ contains opponents with retained comparison outcomes and $|\mathcal{N}_i| > 0$. Comparisons without usable outcomes are excluded.

The margin $p_{ij} - p_{ji}$ is the expected comparison outcome when a win, tie, and loss receive $+1$, 0, and -1, respectively. TIE probability remains in the three-way normalization rather than being redistributed to A and B. Using probabilities instead of hard decisions retains weak preference signals that would otherwise be discarded. The resulting reward lies in $[-1, 1]$ and measures relative preference within a caption group, not absolute quality. The same construction applies to both generator families because it depends only on the caption and rendered audio.

4.1 IN-DOMAIN REWARD-GUIDED OPTIMIZATION

We optimize Audex-2B and Audex-30B-A3B, autoregressive text-to-music models that represent music as interleaved XCodec1 tokens (Ye et al., 2025). For each caption c, candidate i is generated as a codec-token sequence $\mathbf{a}_{c,i}$, decoded to audio $x_{c,i}$, and assigned the reward $r_{c,i}$ from Equation (2). Let θ denote the trainable generator parameters.

CFG-aware policy. Classifier-free guidance (CFG) combines conditional and unconditional predictions to strengthen conditioning without a separate classifier. Originally introduced for diffusion models (Ho & Salimans, 2022), it also applies to autoregressive next-token logits (Sanchez et al., 2024). Following Audex (Kong et al., 2026), we define the guided audio-token policy as

$$\pi_\theta^{\mathrm{cfg}}(\cdot \mid c, \mathbf{a}_{<t}) = \mathrm{softmax}\left(\frac{\gamma \mathbf{z}_{\theta,t}^{c} + (1-\gamma)\mathbf{z}_{\theta,t}^{\emptyset}}{\tau}\right). \tag{3}$$

Here, $\mathbf{z}^{c}_{\theta,t}$ and $\mathbf{z}^{\emptyset}_{\theta,t}$ are logits for caption c and its length-matched null prompt $\emptyset$, evaluated on the same generated prefix. The guidance scale γ controls conditioning strength: $\gamma = 1$ recovers conditional sampling, while $\gamma > 1$ amplifies the conditional–unconditional logit difference. The parameter $\tau > 0$ is the standard language-model sampling temperature. Using conditional-only likelihoods for guided rollouts would introduce a distribution mismatch. We therefore use Equation (3) consistently for rollout generation and current-, old-, and reference-policy likelihoods, with the same γ, τ.

Sequence-level optimization. We optimize the guided policy with GSPO (Zheng et al., 2025), using within-caption standardized rewards as sequence-level advantages. Its loss, $\mathcal{L}_{\mathrm{GSPO}}$, applies clipping to length-normalized sequence likelihood ratios. Null logits are computed from the current model but detached for GSPO, preserving guided probabilities while routing reward gradients through the conditional computation. We take one optimizer step per rollout batch; Appendix C.2.1 details advantage normalization, the objective, and gradient paths.

Guided and null-policy regularization. Because the conditional and null computations share parameters, stopping GSPO gradients through the null branch does not freeze it. We therefore anchor the guided and null policies to a frozen supervised fine-tuning (SFT) reference: $\mathcal{L}^{\mathrm{cfg}}_{\mathrm{reg}}$ constrains the guided sampling distribution, while $\mathcal{L}^{\emptyset}_{\mathrm{reg}}$ constrains the unconditional baseline used by CFG. The total objective is

$$\mathcal{L}(\theta) = \mathcal{L}_{\mathrm{GSPO}}(\theta) + \lambda_{\mathrm{cfg}}\mathcal{L}^{\mathrm{cfg}}_{\mathrm{reg}}(\theta) + \lambda_{\emptyset}\mathcal{L}^{\emptyset}_{\mathrm{reg}}(\theta), \tag{4}$$

with $\lambda_{\mathrm{cfg}} = 0.005$ and $\lambda_{\emptyset} = 0.1$ unless otherwise stated. Both regularizers are sampled, KL-inspired penalties; the null regularizer backpropagates through the current null branch. Appendices C.2.2 and D.2 describe the penalties and training settings.

4.2 Out-of-Domain Reward Transfer

We test reward transfer to ETTA (Lee et al., 2025), whose generated audio is excluded from *DuetJudge* training. The judge remains frozen and supplies the same pairwise rewards as in Equation (2). We adapt DiffusionNFT (Zheng et al., 2026) to optimize ETTA's full generative backbone from completed generations, using within-caption preference weights and forward-process flow matching without storing sampling trajectories or differentiating through the sampler. Rollout generation and training use the same CFG scale, and guided velocity predictions are regularized toward those of the frozen initial ETTA model. Appendices C.3 and D.3 detail reward normalization, the optimization objective, and training settings.

4.3 Main Results

We evaluate reward-guided optimization on Audex-2B and Audex-30B-A3B and reward transfer to ETTA using the non-vocal subset of the Song Describer Dataset (Manco et al., 2023), following ETTA (Lee et al., 2025). For Audex-2B and ETTA, we use the music evaluators introduced in Section 3.2 as individual training-reward baselines. Each evaluator independently scores every rollout, and we discretize its raw scalar score using metric-specific bins to obtain the training reward, without pairwise comparisons.

For automatic evaluation, we report five scores from these evaluators: Audiobox Aesthetics CE and PQ, CMI-RM Musicality and Alignment, and MuQ-MuLan score (denoted MuQ in tables). Table 3 additionally reports two human ratings, OVL and REL, on a 1–5 scale. "Reference Audio" denotes the dataset recordings, while "Base" denotes each generator's initialization without RL.

In-domain reward-guided optimization. *DuetJudge* is the only evaluated reward that improves all five automatic metrics and both human ratings over Audex-2B Base (Table 3). Human OVL and REL rise from 2.697 to 3.147 and from 2.606 to 2.927, respectively. Among the tested rewards, *DuetJudge* achieves the highest MuQ and OVL, while MuQ-MuLan attains the highest REL without improving OVL. In contrast, optimizing Audiobox-CE or CMI-RM Musicality lowers both human ratings despite gains in the corresponding automatic metric, and TuneJury improves human ratings but reduces MuQ. These comparisons support *DuetJudge* as a reward for jointly improving alignment and quality.

Table 3: Reward-guided generation results for Audex-2B, Audex-30B-A3B (in-domain), and ETTA (out-of-domain). ↓ marks scores below the Base; † marks the training-reward metric. *DuetJudge* improves all five automatic metrics and both human ratings over each generator's initialization.

Training reward	Audiobox Aesthetics ↑ (1–10)		CMI-RM ↑ (1–5)		MuQ ↑	Human Evaluation ↑ (1–5)	
	CE	PQ	Musicality	Alignment		OVL	REL
Reference Audio	7.368	7.862	1.883	1.698	0.301	3.765	3.606
In-Domain Reward-Guided Optimization (Base model: Audex-2B (Kong et al., 2026))							
Base (no RL)	6.852	7.257	0.651	1.006	0.209	2.697	2.606
Audiobox-CE	**8.137**†	8.022	0.718	1.025	0.142↓	2.412↓	2.206↓
Audiobox-PQ	7.944	**8.478**†	0.831	1.041	0.151↓	2.912	2.697
CMI-RM Musicality	8.009	8.342	**1.961**†	1.643	0.168↓	2.636↓	2.312↓
CMI-RM Alignment	7.697	7.782	1.900	**2.067**†	0.076↓	2.606↓	2.656
MuQ-MuLan	7.367	7.737	0.793	1.082	0.264†	2.697	**3.030**
TuneJury	7.994	8.205	1.283	1.200	0.145↓	2.971	2.758
***DuetJudge* (ours)**	7.717	8.123	1.303	1.435	**0.296**	**3.147**	2.927
In-Domain Reward-Guided Optimization (Base model: Audex-30B-A3B (Kong et al., 2026))							
Base (no RL)	7.210	7.631	1.029	1.351	0.299	2.818	3.112
***DuetJudge* (ours)**	**7.743**	**8.163**	**1.634**	**1.686**	**0.350**	**3.545**	**3.375**
Out-of-Domain Reward Transfer (Base model: ETTA (Lee et al., 2025))							
Base (no RL)	7.075	7.549	1.126	1.358	0.279	2.879	3.100
Audiobox-CE	**7.342**†	7.822	1.287	1.431	0.287	3.319	2.842↓
Audiobox-PQ	7.108	**8.142**†	1.155	1.227↓	0.283	**3.519**	2.842↓
CMI-RM Musicality	6.984↓	7.760	1.276†	1.380	0.293	3.039	2.947↓
CMI-RM Alignment	7.256	7.818	**1.359**	**1.489**†	0.295	3.279	3.211
MuQ-MuLan	6.866↓	7.714	1.049↓	1.233↓	**0.348**†	3.319	**3.474**
TuneJury	7.244	7.772	1.262	1.414	0.291	3.239	3.105
***DuetJudge* (ours)**	7.143	7.724	1.268	1.419	0.311	3.399	3.316

Reward optimization across model scales. The same frozen judge improves Audex-30B-A3B on all five automatic metrics and both human ratings. After RL, the 30B model also outperforms the optimized 2B model on every reported metric. Reward-guided optimization therefore complements model scaling, benefiting both the smaller generator and the stronger initialization.

Out-of-domain reward transfer. The same frozen *DuetJudge* improves ETTA on all five automatic metrics and both human ratings, despite never having been trained on ETTA-generated audio. It achieves the second-highest OVL and REL among the tested rewards. Audiobox-PQ leads in OVL but lowers REL below Base, whereas MuQ-MuLan leads in REL and improves OVL despite declines in three automatic metrics. CMI-RM Alignment and TuneJury also improve all seven scores, but *DuetJudge* yields higher OVL, REL, and MuQ than both. These results support reward transfer to an unseen generator, with gains reflected in both automatic and human evaluations.

4.4 Ablation Studies on Audex-2B

Pairwise versus pointwise rewards. Table 4 compares three ways of using preference feedback: DPO trained directly on the existing non-tie judgment pairs, GSPO with a pointwise *DuetJudge*-BTRM that converts pairwise judgments into scalar rewards via Bradley–Terry modeling (Bradley & Terry, 1952), and GSPO with direct pairwise *DuetJudge* rewards. Under GSPO, the direct pairwise judge improves all five automatic metrics over BTRM, increasing CMI-RM Musicality from 0.998 to 1.303 and Alignment from 1.229 to 1.435. DPO also improves all five metrics over Base, but switching from DPO to GSPO with BTRM lowers both alignment scores despite improving the quality scores. These results suggest that preserving pairwise feedback during online policy optimization is more effective than either reducing it to pointwise scalar rewards or using it only for offline optimization.

Tie supervision and swap augmentation. Table 5 separates judge-training changes from reward-construction changes. Following prior work on tie-aware reward learning and LLM alignment (Liu et al., 2024; Guo et al., 2025; Chen et al., 2025), we retain explicit Tie supervision for comparisons without a meaningful preference. With order-averaged probability rewards, including tie-labeled pairs raises CMI-RM Musicality from 1.133 to 1.303 and Alignment from 1.315 to 1.435, although

Table 4: Pointwise versus pairwise rewards under GSPO, with DPO as an additional comparison, for Audex-2B. *DuetJudge*-BTRM assigns a scalar score to each caption–audio pair using Bradley–Terry reward modeling. DPO uses existing non-tie judgment pairs from Section 2.

Optimization	Reward model	Audiobox Aesthetics ↑ (1–10)		CMI-RM ↑ (1–5)		MuQ ↑
		CE	PQ	Musicality	Alignment	
Base (no RL)	–	6.852	7.257	0.651	1.006	0.209
DPO	–	7.292	7.701	0.987	1.325	0.288
GSPO	*DuetJudge*-BTRM	7.698	8.035	0.998	1.229	0.269
GSPO	*DuetJudge*	**7.717**	**8.123**	**1.303**	**1.435**	**0.296**

Audiobox-CE decreases slightly. Separately, swap augmentation, which adds label-consistent reversed training pairs, improves all five automatic metrics under the same reward construction.

Table 5: Core design ablations for Audex-2B RL. The full recipe combines tie-supervised, swap-augmented judge training with order-averaged probability-margin rewards.

Configuration	Audiobox Aesthetics ↑ (1–10)		CMI-RM ↑ (1–5)		MuQ ↑
	CE	PQ	Musicality	Alignment	
Full recipe (default)	7.717	**8.123**	**1.303**	**1.435**	**0.296**
Judge-training changes					
w/o tie-labeled pairs	**7.771**	8.119	1.133	1.315	0.278
w/o swap augmentation	7.670	8.019	1.054	1.252	0.289
Reward-construction changes					
w/o order averaging	7.747	8.020	1.248	1.406	0.295
Hard-decision rewards	7.588	7.939	1.165	1.368	0.283

Probability margins and order averaging. For the swap-trained judge, probability margins outperform hard decisions on all five automatic metrics (Table 5). Averaging both presentation orders additionally raises Audiobox-PQ from 8.020 to 8.123 and CMI-RM Alignment from 1.406 to 1.435, with a small decrease in Audiobox-CE. These results support combining swap-augmented training with order-averaged probability rewards. The advantage of probability margins is not uniform without swap augmentation.

Appendix E complements these core ablations with the full candidate-order results, comparison-set experiments, and hyperparameter sweeps. In particular, the comparison-set results show that intra-group comparisons provide effective rewards without requiring real-audio references in this setting.

5 Conclusion

We introduced *DuetJudge*, a caption-conditioned pairwise reward model for instrumental music built around an explicit alignment-first, quality-second criterion: meaningful differences in caption alignment determine preference, while perceptual quality distinguishes comparably aligned candidates. Through complementary supervision from holistic preferences, fine-grained alignment comparisons, and controlled quality comparisons, *DuetJudge* learns to jointly evaluate two candidates and predict A, B, or TIE rather than assign independent scores. We convert its pairwise preference probabilities into within-caption rewards for reinforcement learning, improving both caption alignment and perceptual quality in Audex-2B and Audex-30B-A3B through CFG-aware policy optimization. The same frozen judge also improves diffusion-based ETTA, whose outputs were excluded from judge training, supporting reward transfer to an unseen generator. Together, these results demonstrate the practical value of making preference priorities explicit in music reward modeling: criterion-driven pairwise judgments provide effective optimization signals across model scales and architectures, helping generators better fulfill user requests while improving listening quality.

AI USE STATEMENT

We used AI tools for language refinement and grammar checks during paper drafting. We used AI tools to annotate synthetic data, generate synthetic audio, and filter low-quality data in this study, similar to many other LLM and Audio LLM studies. The authors take full responsibility for the research design, experimental results, interpretations, and final content of this paper.

ETHICS STATEMENT

This work studies reward modeling and reinforcement learning for instrumental music generation. Our alignment-first, quality-second criterion is a task-specific preference rule and does not represent a universal measure of artistic value. The learned judge may inherit biases from its source data and annotations, and optimizing against it may favor musical styles already well represented in those sources. Automated judgments should therefore complement, not replace, human listening and creative judgment.

REFERENCES

Nemo rl: A scalable and efficient post-training library. https://github.com/NVIDIA-NeMo/RL, 2025. GitHub repository.

Pablo Alonso-Jiménez, Dmitry Bogdanov, Jordi Pons, and Xavier Serra. Tensorflow audio models in Essentia. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Jinghong Chen, Guangyu Yang, Weizhe Lin, Jingbiao Mei, Chenxu Lyu, and Bill Byrne. On extending direct preference optimization to accommodate ties. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Geoffrey Cideron, Sertan Girgin, Mauro Verzetti, Damien Vincent, Matej Kastelic, Zalán Borsos, Brian McWilliams, Victor Ungureanu, Olivier Bachem, Olivier Pietquin, et al. Musicrl: Aligning music generation to human preferences. *arXiv preprint arXiv:2402.04229*, 2024.

Hadrien Foroughmand and Geoffroy Peeters. Deep-rhythm for tempo estimation and rhythm pattern recognition. In *International Society for Music Information Retrieval (ISMIR)*, 2019.

Sreyan Ghosh, Arushi Goel, Jaehyeon Kim, Sonal Kumar, Zhifeng Kong, Sang-gil Lee, Chao-Han Yang, Ramani Duraiswami, Dinesh Manocha, Rafael Valle, et al. Audio flamingo 3: Advancing audio intelligence with fully open large audio language models. *Advances in Neural Information Processing Systems*, 38:41819–41886, 2026.

Yuxiang Guo, Lu Yin, Bo Jiang, and Jiaqi Zhang. TODO: Enhancing LLM alignment with ternary preferences. In *International Conference on Learning Representations*, 2025.

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

Yonghyun Kim, Junwon Lee, Haiwen Xia, Yinghao Ma, Junghyun Koo, Koichi Saito, Yuki Mitsufuji, and Chris Donahue. Tunejury: An open metric for improving music generation preference alignment. *arXiv preprint arXiv:2606.17006*, 2026.

Yuexuan Kong, Gabriel Meseguer-Brocal, Vincent Lostanlen, Mathieu Lagrange, and Romain Hennequin. S-key: Self-supervised learning of major and minor keys from audio. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025.

Zhifeng Kong, Sang-gil Lee, Jaehyeon Kim, Boxin Wang, Zihan Liu, Sungwon Kim, Yang Chen, Arushi Goel, Rajarshi Roy, Wenliang Dai, et al. Unified audio intelligence without regressing on text intelligence. *arXiv preprint arXiv:2607.05196*, 2026.

Sang-gil Lee, Zhifeng Kong, Arushi Goel, Sungwon Kim, Rafael Valle, and Bryan Catanzaro. ETTA: Elucidating the design space of text-to-audio models. In *International Conference on Machine Learning*, 2025.

Jinsong Liu, Dongdong Ge, and Ruihao Zhu. Reward learning from preference with ties. *arXiv preprint arXiv:2410.05328*, 2024.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *Machine Intelligence Research*, 22 (4):730–751, 2025.

Yinghao Ma, Haiwen Xia, Hewei Gao, Weixiong Chen, Yuxin Ye, Yuchen Yang, Sungkyun Chang, Mingshuo Ding, Yizhi Li, Ruibin Yuan, et al. Cmi-rewardbench: Evaluating music reward models with compositional multimodal instruction. *arXiv preprint arXiv:2603.00610*, 2026.

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 at NeurIPS 2023*, 2023.

Adrian C. North and David J. Hargreaves. Responses to music in aerobic exercise and yogic relaxation classes. *British Journal of Psychology*, 87(4):535–547, 1996. doi: 10.1111/j.2044-8295.1996.tb02607.x.

Guillaume Sanchez, Alexander Spangher, Honglu Fan, Elad Levi, and Stella Biderman. Stay on topic with classifier-free guidance. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 43197–43234. PMLR, 2024.

Thomas Schäfer. The goals and effects of music listening and their relationship to the strength of music preference. *PLOS ONE*, 11(3):e0151634, 2016. doi: 10.1371/journal.pone.0151634.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019.

Paul Slovic. The construction of preference. *American Psychologist*, 50(5):364–371, 1995. doi: 10.1037/0003-066X.50.5.364.

Andros Tjandra, Yi-Chiao Wu, Baishan Guo, John Hoffman, Brian Ellis, Apoorv Vyas, Bowen Shi, Sanyuan Chen, Matt Le, Nick Zacharov, et al. Meta audiobox aesthetics: Unified automatic quality assessment for speech, music, and sound. *arXiv preprint arXiv:2502.05139*, 2025.

Zhen Ye, Peiwen Sun, Jiahe Lei, Hongzhan Lin, Xu Tan, Zheqi Dai, Qiuqiang Kong, Jianyi Chen, Jiahao Pan, Qifeng Liu, et al. Codec does matter: Exploring the semantic shortcoming of codec for audio language model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 25697–25705, 2025.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025.

Kaiwen Zheng, Huayu Chen, Haotian Ye, Haoxiang Wang, Qinsheng Zhang, Kai Jiang, Hang Su, Stefano Ermon, Jun Zhu, and Ming-Yu Liu. DiffusionNFT: Online diffusion reinforcement with forward process. In *International Conference on Learning Representations*, 2026.

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. *IEEE Transactions on Audio, Speech and Language Processing*, 2025.

A JUDGMENT DATA CONSTRUCTION DETAILS

This appendix provides details omitted from Section 2. The source pools below support holistic and perceptual-quality comparisons; fine-grained alignment uses separate attribute-annotated pools. All candidates are 10-second clips. Besides real data from academic sources, we also draw on synthetic data generated by models including Nemotron-Labs-Audex-2B and Nemotron-Labs-Audex-30B-A3B (Kong et al., 2026), referred to as Audex-2B and Audex-30B-A3B, respectively, throughout the main body and the descriptions below.

A.1 HOLISTIC PREFERENCE

The holistic preference data in Section 2.1 contain real–real, real–synthetic, and synthetic–synthetic comparisons (Table 6). Each sample is a caption-conditioned comparison.

Table 6: Holistic preference data before candidate-order reversal. Real–synthetic and synthetic–synthetic counts are aggregated across subsets.

Pair type	Comparison subset	# Samples	Audio hours	Audio tokens
Real–real	Same-recording excerpts (ties)	14,152	79	7 M
31,916 total	Retrieved similar recording	13,166	73	7 M
	Similar-caption crosswise	4,598	25	2 M
Real–synthetic	Same-caption pairs	50,577	281	26 M
Synthetic–synthetic	Cross-model / cross-seed	80,834	450	40 M
Total		163,327	908	82 M

A.1.1 CANDIDATE PRE-SELECTION AND EXPERT ANNOTATION

Metric-based candidate preselection. For metric-based preselection, we use MuQ-MuLan (Zhu et al., 2025) for caption–audio alignment and Audiobox-PQ (Tjandra et al., 2025) for perceptual quality. Metrics are used only for candidate selection, not for label construction.

Expert annotation. Music experts compare each selected pair under the shared caption and assign a label in $\{\mathrm{A}, \mathrm{B}, \textsc{Tie}\}$ using the alignment-first, quality-second criterion. Candidate order is randomized. Within each construction family, we retain only pairs whose expert labels agree with the intended preference direction or tie relation.

A.1.2 REAL–REAL COMPARISONS

Same-recording excerpts. For source excerpts longer than 25 seconds, we randomly sample two 10-second segments from the same recording and pair them under their shared caption. We retain only pairs that experts label as TIE.

Retrieved similar recordings. Given a caption–audio pair (c_A, x_A), we search real excerpts from the same source collection, excluding the original recording. We retrieve the five candidates x_B with the highest alignment similarities to c_A. One or two retrieved candidates are sampled to form comparisons with x_A as the intended winner, after which candidate positions are randomized.

Similar-caption crosswise comparisons. We also use c_B, the caption associated with a retrieved candidate x_B. Both original caption–audio pairs, (c_A, x_A) and (c_B, x_B), must have alignment scores of at least 0.45; the crossed pairs, (c_A, x_B) and (c_B, x_A), must score 0.1–0.2 lower than their respective originals. With the same top-five retrieval and random sampling procedure, each audio pair yields two comparisons: (c_A, x_A, x_B) favors x_A, whereas (c_B, x_A, x_B) favors x_B. We deduplicate by ordered winner–loser identity.

A.1.3 REAL–SYNTHETIC COMPARISONS

We pair real excerpts with outputs generated from their captions, including those from Audex-2B and Audex-30B-A3B. We retain real-preferred, generated-preferred, and tie pairs according to the expert annotation.

A.1.4 SYNTHETIC–SYNTHETIC COMPARISONS

We construct cross-model comparisons by pairing outputs from different models under the same caption. We also form same-model comparisons between independent Audex-2B outputs, using different random seeds while keeping the caption and default inference configuration fixed.

A.2 FINE-GRAINED ALIGNMENT

The fine-grained data in Section 2.2 use attribute-annotated real recordings. Each caption specifies one attribute, and the two recordings may differ in other properties. Reference annotations determine the labels; auxiliary estimators and classifiers verify their local consistency within the selected 10-second clips.

Non-ties are described as $(c, x_A, x_B, \mathrm{A})$, where x_A matches the caption and x_B is a verified negative; the matching side is balanced in stored examples. Ties, where included, pair two verified matches. Pairs stay within the same source collection to reduce recording-domain shortcuts. Table 7 reports the number of comparisons for each task.

Table 7: Fine-grained alignment data by task. Counts before candidate-order reversal.

	# Samples	Audio hours	Audio tokens
Tempo	23,747	132	12 M
Instrument	4,322	24	2 M
Key	15,675	87	8 M
Chord	3,358	19	2 M
Genre	3,865	21	2 M
Mood	14,185	79	7 M
Total	65,152	362	33 M

A.2.1 TEMPO

Source labels and verification. The tempo pool includes reference tempi obtained from expert annotations and manually verified beat annotations. We consolidate these annotations into a single reference tempo for each excerpt and retain only excerpts with reliable reference tempi in the range of 65–185 BPM.

Numerically different tempi can correspond to plausible metrical levels of the same recording. To avoid treating these alternatives as negatives, we define

$$\mathcal{E}(s) = \mathcal{M}(s) \cup \left\{ \alpha b_s : \alpha \in \left\{ \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{3}{4}, \frac{4}{3}, \frac{3}{2}, 2, 3 \right\} \right\},$$

where b_s is the reference tempo and $\mathcal{M}(s)$ contains all annotated metrical levels, including those below the agreement threshold used for reference selection. The set $\mathcal{M}(s)$ is empty when no multi-level annotations are available. Positive selection uses the stricter reference tempo, whereas $\mathcal{E}(s)$ prevents plausible alternatives from being treated as negatives.

We extract non-overlapping 10-second clips, remove silent excerpts, and verify local tempo consistency with DeepRhythm (Foroughmand & Peeters, 2019). A positive anchor requires an estimate within 5% of its recording's reference tempo; estimates are also checked against $\mathcal{E}(s)$ for consistency with plausible alternatives.

Pair construction. The caption states the rounded reference BPM of x_A, for example, "a piece of music at 120 BPM." A negative x_B comes from a different recording in the same split. Its reference tempo must differ from the caption BPM by a larger-to-smaller ratio of at least 1.2, and neither its plausible metrical tempi nor its DeepRhythm estimate may fall within 8% of the target. Ratios in $[1.2, 1.45]$ define *near* negatives, and ratios above 1.45 define *far* negatives. For the multi-level annotation pool, pairing preferentially stays within genre to reduce genre and timbre shortcuts.

Ties pair independently verified anchors from different recordings whose reference tempi round to the same caption BPM.

A.2.2 INSTRUMENT

Source labels and verification. We construct separate lead-instrument and solo-instrument pools. The lead-instrument pool uses 10-second excerpts from multi-instrument recordings with human predominant-instrument annotations, excluding voice-labeled clips. For negative selection, we aggregate instrument labels across all annotated excerpts of each song, including those excluded from the positive pool. The solo-instrument pool uses 10-second excerpts with sparse instrument-presence annotations. We screen these excerpts for solo performances, excluding vocal, mixed-instrument, and ambiguous examples while preserving the original instrument labels.

Pair construction. Lead-instrument captions specify the predominant instrument, for example, "a piece of music led by the violin" or "the saxophone carries the main melody." The target must appear in x_A's excerpt-level labels and be absent from the song-level label union of x_B.

Solo-instrument captions describe exclusive presence, for example, "a solo piano piece" or "a track featuring only the guitar." Lead-oriented templates are also valid for solo performances, but non-melodic instruments such as drums and cymbals use only non-melodic templates. Here, x_A is a curated solo of instrument X, and x_B is a curated solo of another instrument $Y \neq X$. We reject negatives whose original annotations provide conflicting evidence of X's presence.

Negatives are *hard* within an instrument family and *easy* across families. The lead-instrument grouping covers strings, acoustic and electric guitars, keyboards, and winds; the solo-instrument grouping covers bowed strings, plucked strings, keyboards, winds, and percussion. All 4,322 comparisons are non-ties, comprising 2,006 lead-instrument and 2,316 solo-instrument comparisons.

A.2.3 KEY

Source labels and verification. Reference labels are normalized to the 24 major and minor keys with canonical enharmonic pitch classes, for example, D♯ and E♭. Ambiguous or unmappable labels are discarded.

We apply S-key (Kong et al., 2025) to 10-second windows with a 5-second hop to verify recording-level labels locally. A recording is retained only when more than half its window predictions agree with its reference key; a retained window additionally requires an agreeing prediction with confidence at least 0.7.

Pair construction. The caption specifies the key of x_A, for example, "a piece of music in C major." Different templates and enharmonic spellings may be used, while the underlying key representation is kept canonical. A negative x_B is required to have a different verified key from x_A. We vary the similarity of negative pairs according to the number of diatonic pitch classes shared by the two keys, with fewer shared pitch classes corresponding to more distinct negatives and six shared pitch classes to closely related negatives. Parallel major and minor keys are also included as closely related negatives because they share the same tonic but differ in mode. Relative major and minor keys are excluded because they have identical diatonic pitch-class sets and can be ambiguous in short excerpts when tonic cues are weak. For tie samples, x_B is drawn from a different recording with the same key as x_A.

A.2.4 CHORD

Source labels and verification. Chord labels are reduced to major and minor triads. Qualities containing a major triad, including dominant and major-seventh extensions, map to major chords;

qualities containing a minor triad map to minor chords. We ignore inversions and canonicalize enharmonic roots.

Ten-second windows start at annotated chord boundaries. Boundary fragments shorter than 40 ms are ignored to avoid floating-point artifacts. Each retained chord must be audible for at least 0.8 seconds, no chord may occupy more than 4.5 seconds, and annotation gaps cannot exceed 0.05 seconds. Sequences must be strictly periodic with fundamental period $p \in \{3, 4, 5\}$, contain at least three distinct chords, and include a full cycle followed by evidence of repetition.

Pair construction. The caption names the first complete cycle heard from the beginning of x_A, for example, "a song with an Am-F-C-G chord progression." Progressions are equivalent up to cyclic rotation because loops have no privileged starting point. Pairing uses canonical rotations, while captions retain the heard rotation. We accept a negative x_B only when no rotation of the target progression occurs contiguously in its loop and the converse containment test also passes. This excludes rotational equivalents and embedded sub-progressions.

To construct challenging negatives, we include pairs drawn from different sections of the same recording, as well as pairs whose progressions have equal periods and differ in exactly one aligned chord, use the same chord set in a non-rotationally equivalent order, or share the same interval-and-quality pattern under different absolute roots. Among the remaining valid negatives, we distinguish cases sharing at least two chords from those sharing at most one. Except for pairs drawn from different sections of the same song, negative candidates come from different recordings. Ties are formed across different songs when both clips contain the same rotation-equivalent progression.

A.2.5 GENRE

Source labels and verification. We encode genre families and parent–child relationships to distinguish hierarchically related labels from genuinely contradictory ones. For example, ambient, house, techno, trance, and drum and bass belong to the broader electronic family, with several of them also treated as descendants of electronic or dance. Coarse-grained or overlapping labels—including folk-country, funk/soul/R&B, alternative, and disco—are excluded from the target label set but retained when screening negative candidates. The final target set consists of ambient, classical, dance, electronic, hip-hop, jazz, metal, pop, rock, and trance.

Pair construction. The caption specifies x_A's verified genre, for example, "a jazz track" or "music in the style of metal." A negative x_B must have recording-level labels containing neither the target genre nor any of its descendants.

Negative pairs are constructed either across genre families or between safely distinguishable sibling genres, such as ambient versus techno or jazz versus blues. Parents are never contrasted with their descendants; for example, electronic is not paired against one of its own subgenres as a negative. Ties pair independently verified positives for the target genre from different recordings.

A.2.6 MOOD

Source labels and verification. We consider seven attributes: acoustic, aggressive, electronic, happy, party, relaxed, and sad. We extract 10-second windows with a 5-second hop, excluding the first and last 5% of each recording, and verify them using attribute-specific EffNet classifiers (Alonso-Jiménez et al., 2020). We calibrate confidence thresholds separately for each attribute and retain only windows whose source labels agree with the classifier scores: positive examples carry the target label and score above the positive threshold, whereas negative examples carry an explicit negative label and score below the negative threshold.

Pair construction. Captions describe a single target attribute using varied templates and synonyms, such as *cheerful*, *upbeat*, or *joyful* for happy; *melancholic* or *sorrowful* for sad; and *calm*, *mellow*, or *soothing* for relaxed. For non-tie pairs, x_A is a verified positive for the target attribute, while x_B has an explicit negative label for that attribute and a classifier score below its negative threshold. Among these eligible negatives, some are also verified positives for a predefined contrasting attribute: happy versus sad, relaxed versus aggressive, party versus relaxed or sad, and acoustic versus electronic. The remaining negatives are drawn from other clips that meet the same label and score requirements for

the target attribute. For ties, we instead pair independently verified positives for the target attribute from different recordings.

A.3 Perceptual Quality

The quality data in Section 2.3 contain 59,239 comparisons between variants of the same source clip (Table 8). We assign the preferred label to the variant with no additional degradation or lower severity within the same degradation family; no ties are included.

Table 8: Perceptual-quality data by task and audio source. Each sample compares two variants of the same source clip under a shared caption, with no tie labels. Counts before candidate-order reversal.

Task	**Audio source**	**# Samples**	**Audio hours**	**Audio tokens**
Normal vs. degraded	Real	19,739	109	10 M
Normal vs. degraded	Generated	9,873	55	5 M
Light vs. heavy	Real	19,755	110	10 M
Light vs. heavy	Generated	9,872	55	5 M
Total		59,239	329	30 M

A.3.1 Source Screening and Shared Preprocessing

We require source Audiobox-PQ scores of at least 6.0 to reduce cases where an added degradation masks existing artifacts and improves perceived quality. Scores determine source eligibility only; degraded candidates are not rescored. Both candidates undergo the same preprocessing to 16-kHz mono, 16-bit PCM waveforms, with peaks limited to 0.999, avoiding sample-rate or channel-format cues to the preferred side.

A.3.2 Normal-versus-Degraded Comparisons

One candidate receives only shared preprocessing, while the other additionally receives one of the 15 presets in Table 9. We assign one preset per source clip and prefer the candidate without additional degradation. Here, "normal" does not mean artifact-free audio. The retained normal-versus-degraded subset contains 19,739 real-source and 9,873 generated-source comparisons.

A.3.3 Light-versus-Heavy Comparisons

Each source clip is assigned one of the eight degradation families in Table 10. We sample two severity levels, render only those variants, and prefer the lighter one. Comparisons never cross degradation families. For four-level ladders, levels one, two, or three steps apart are sampled with probabilities 50%, 30%, and 20%; three-level ladders use distances of one and two with probabilities 60% and 40%. Adjacent levels provide the hardest intended comparisons. This task contributes 19,755 real and 9,872 generated comparisons.

A.3.4 Degradation Operators

Low-pass and high-pass filtering use second-order Butterworth filters in second-order-section form; band-pass effects compose both. MP3 uses constant-bitrate LAME encoding through libsndfile. Opus and GSM use 8-kHz coding, with effective Opus rates of approximately 25, 12, and 9.4 kbps for the nominal settings in Table 10. MP3, Opus, and GSM outputs are cross-correlation-aligned to remove codec delay and restore the input sample count.

White-noise amplitude is set from measured signal power and the target SNR; compound pink-noise presets use fixed peak amplitudes. Bit crushing applies μ-law companding ($\mu = 100$) before quantization. Clipping peak-normalizes the waveform, applies the specified drive and hard clipping, and restores the original RMS to reduce loudness shortcuts. Packet loss zeros selected 40-ms packets with 3-ms cosine boundary ramps.

Table 9: The 15 presets used for normal-versus-degraded comparisons. Presets cover different artifact types and are not treated as a globally ordered quality scale or directly compared against one another.

Preset	Processing chain	Intended perceptual effect
`lp5k`	Low-pass at 5 kHz	Mild loss of high-frequency detail
`lp35n`	Low-pass at 3.5 kHz; add pink noise with peak amplitude 0.02	Reduced brightness with audible background noise
`mild`	Low-pass at 5.8 kHz; MP3 at 32 kbps	Mild lossy-coding artifacts
`medium`	Low-pass at 3.8 kHz; resample to 8 kHz; MP3 at 16 kbps	Stronger bandwidth loss and underwater-like coding artifacts
`phone`	Band-pass from 300 to 3,400 Hz; GSM encode–decode	Telephone-bandwidth audio
`crush7`	Low-pass at 4.2 kHz; 7-bit crushing with mixture weight 0.6; MP3 at 24 kbps	Moderate digital granularity
`muffled`	Low-pass at 2.5 kHz; resample to 8 kHz; MP3 at 8 kbps	Severe muffling and compression artifacts
`radio`	Band-pass from 400 to 1,800 Hz; apply $+8$ dB gain and hard clipping; add pink noise with peak amplitude 0.03	AM-radio-like distortion
`crush4`	Low-pass at 3 kHz; 4-bit crushing with mixture weight 1.0; MP3 at 16 kbps	Strong digital quantization artifacts
`warbly`	Resample to 8 kHz; Opus at approximately 9 kbps	Warbling or robotic coding artifacts
`snr20`	Add white noise at 20-dB SNR	Mild additive noise
`snr10`	Add white noise at 10-dB SNR	Strong additive noise
`clip`	Peak-normalize; apply drive factor $\times 4$ and hard clipping; restore the original RMS	Nonlinear clipping distortion
`plr`	Remove 10% of 40-ms packets with 3-ms boundary ramps	Intermittent dropouts
`xcodec`	XCodec1 encode–decode at 2.0 kbps using four codebooks	Neural-codec reconstruction artifacts

Table 10: Ordered degradation families used for light-versus-heavy comparisons.

Family	Severity ladder (lighter $\rightarrow$ heavier)
Low-pass	6.0 $\rightarrow$ 4.5 $\rightarrow$ 3.0 $\rightarrow$ 2.0 kHz
Noise	30 $\rightarrow$ 20 $\rightarrow$ 10 $\rightarrow$ 0 dB SNR
MP3	48 $\rightarrow$ 32 $\rightarrow$ 16 $\rightarrow$ 8 kbps (16 kHz)
Opus	24 $\rightarrow$ 12 $\rightarrow$ 9 kbps (8 kHz)
Bit crushing	8 $\rightarrow$ 6 $\rightarrow$ 4 $\rightarrow$ 3 bits
Clipping	$\times 2 \rightarrow \times 4 \rightarrow \times 8$ drive
Packet loss	5% $\rightarrow$ 10% $\rightarrow$ 20% (40-ms packets)
XCodec	4.0 $\rightarrow$ 2.0 $\rightarrow$ 1.0 $\rightarrow$ 0.5 kbps (8 / 4 / 2 / 1 codebooks)

B *DuetJudge* Training and Implementation Details

B.1 Audex Backbone and Pairwise-Judge Adaptation

Backbone. We build *DuetJudge* on Audex-2B (Kong et al., 2026), an audio-language model supporting both audio understanding and generation. Its language backbone is a dense decoder-only Transformer with 28 layers, a hidden dimension of 2,048, and a feed-forward dimension of 9,216. It uses grouped-query attention with 16 query heads and 8 key–value heads, RMSNorm, squared-ReLU activations, and rotary positional embeddings with a base of 10^8. Linear layers are bias-free, and input and output embeddings are untied.

Audio representation. We use Audex's AF-Whisper audio encoder, inherited from Audio Flamingo 3 (Ghosh et al., 2026). Each 10-second candidate is resampled to 16 kHz and zero-padded to a 30-second encoder window, yielding 750 continuous feature vectors of dimension 1,280 at 25 Hz. A two-layer MLP adapter projects these features into the language model's 2,048-dimensional embedding space. The two candidates are encoded separately and presented together with the caption and comparison instructions in a shared context.

Judge adaptation. We retain the pretrained architecture and language-model output head without adding a scalar reward head. During pairwise-judgment specialization, we fine-tune the full language model, including its token embeddings, and the audio projection adapter, while keeping AF-Whisper frozen.

B.2 Pairwise Input Format and Decision-Only Supervision

All three judgment-data families share the comparison prompt shown below. The placeholders `<sound-1>` and `<sound-2>` mark the positions where the projected audio features of x_A and x_B, respectively, are inserted.

> Given the caption: {caption}
>
> Compare Audio A <sound-1> and Audio B <sound-2> as 10-second instrumental caption-conditioned music candidates.
>
> Judge primarily by caption alignment: genre/style, timbre or instrumentation, mood, rhythm/groove, energy, broad tempo, and overall musical behavior. Treat exact BPM, key, chords, and narrowly specified instruments as soft evidence; do not invent them if uncertain. Penalize prominent lead vocals or lyrical singing unless the caption allows vocal-like textures. Use musical quality only as a tiebreaker when alignment is close, considering coherence, timing, mix clarity, artifacts, and musical intentionality. Choose Tie if alignment and quality are comparable or evidence is insufficient.
>
> Output only valid JSON:
> `{"decision":"<A|B|Tie>"}`

For each pairwise example, the assistant target is a JSON object containing only the `decision` field, with value `A`, `B`, or `Tie`, and no reasoning trace. We apply autoregressive cross-entropy to the assistant response, including its formatting and end-of-turn tokens, while masking all system, user, and audio-input positions.

B.3 Two-Stage Training

Stage I: Music Understanding. Stage I uses audio- and text-conditioned text generation without pairwise-preference supervision. The training mixture focuses on music understanding, covering question answering, captioning, chord recognition, tempo and key estimation, and lyrics- and MIDI-related tasks. Supplementary tasks include general audio understanding, speech recognition and translation, and text-only instruction following.

Stage II: Pairwise-Judgment Specialization. Starting from the Stage-I checkpoint, we fine-tune *DuetJudge* on the judgment data constructed in Appendix A, using the decision-only supervision described in Appendix B.2. Stage II is substantially smaller than Stage I and specializes the model's music-understanding capabilities into an alignment-first, quality-second comparison policy.

Judgment-data mixture. Rather than sampling examples in proportion to raw dataset size, we choose mixture weights according to the comparison behavior each component is intended to reinforce. Holistic comparisons anchor broad preference judgments, while same-caption synthetic comparisons receive additional emphasis because they closely match the within-prompt rollout comparisons encountered during downstream preference optimization. Fine-grained examples retain sensitivity to explicit caption constraints and enforce the intended priority of alignment over perceptual quality.

We sample perceptual-quality supervision less aggressively than alignment-focused supervision so that easily detected production artifacts do not dominate subtler semantic distinctions. Within the quality component, we place relatively greater emphasis on generated-domain examples, which more closely match the audio distribution encountered during downstream reward optimization.

Tie calibration. The explicit Tie option allows the judge to abstain when the difference between candidates falls below its effective comparison resolution. Forced choices can become order-sensitive and introduce noisy preference signals into downstream optimization. We therefore emphasize Tie supervision in same-caption synthetic and fine-grained comparisons, where genuine ambiguity is more likely, rather than distributing it uniformly across comparison families. Non-tie supervision remains dominant to avoid overproducing ties on informative within-caption comparisons. The overall non-tie:tie ratio after weighting is 10:1.

Candidate-order augmentation. Each comparison is included in both candidate orders with equal sampling weight, exchanging the A/B labels and leaving TIE unchanged.

Capability retention. During Stage II, we interleave pairwise judgment data with non-comparison audio- and music-understanding examples. These examples ask the model to interpret a single audio clip and produce a textual response, rather than compare two candidates and predict A, B, or TIE. We additionally include music-theory question answering and general text instruction data. This continued supervision is intended to retain the listening, music–language understanding, and instruction-following capabilities underlying pairwise judgment while the model specializes in preference prediction.

C POLICY OPTIMIZATION DETAILS

This appendix supplements Section 4 with reward-computation details and the Audex and ETTA update objectives. We use the candidate reward defined in Equation (2) throughout. Hyperparameters and computational settings are reported in Appendix D.

C.1 REWARD COMPUTATION AND PROPERTIES

Order averaging and reward properties. Using the decision probabilities defined in Section 4, write $\mathbf{q}_{ij} = q_\phi(\cdot \mid c, x_i, x_j)$ and let $S(q^A, q^B, q^{\text{TIE}}) = (q^B, q^A, q^{\text{TIE}})$ exchange the presentation slots. The order-averaged distribution is

$$\overline{\mathbf{q}}_{ij} = \frac{1}{2}\left[\mathbf{q}_{ij} + S(\mathbf{q}_{ji})\right]. \tag{5}$$

Its A and B components are the main-text probabilities p_{ij} and p_{ji}, respectively, so the margin is

$$m_{ij} = p_{ij} - p_{ji} = \overline{q}^A_{ij} - \overline{q}^B_{ij}. \tag{6}$$

Equation (2) averages these margins over retained opponents. The margins satisfy $m_{ji} = -m_{ij}$ and $|m_{ij}| \le 1 - \overline{q}^{\text{TIE}}_{ij}$. Under complete all-pairs comparison, every candidate has $G-1$ opponents, and the group rewards satisfy the zero-sum property:

$$\sum_{i=1}^{G} r_i = \frac{1}{G-1} \sum_{1 \le i < j \le G} (m_{ij} + m_{ji}) = 0. \tag{7}$$

Decision-probability extraction for Audex. For each presentation order, the reward implementation greedily generates a decision-only response and locates the first label-bearing position after the `decision` key within the first 128 generated steps, skipping a standalone quote or empty token when necessary. We apply vocabulary softmax at that position, retain the 50 highest-probability tokens, and sum their probabilities into A, B, and TIE buckets according to whether their decoded strings begin with A, B, or T after removing surrounding whitespace and leading quotation marks. Renormalizing the three masses gives $q_\phi(\cdot \mid c, x_i, x_j)$. These are decision-position estimates, not complete-label sequence likelihoods; the TIE estimate uses T-prefix token mass.

Within-caption statistics. For a caption group in which all G candidate rewards are defined, both updates use

$$\mu_c = \frac{1}{G}\sum_{j=1}^{G} r_{c,j}, \qquad \sigma_c = \text{Std}\left(\{r_{c,j}\}_{j=1}^{G}\right). \tag{8}$$

The Audex and ETTA objectives below transform these shared rewards into sequence advantages and preference weights, respectively.

C.2 IN-DOMAIN OPTIMIZATION: AUDEX

C.2.1 CFG-AWARE SEQUENCE OPTIMIZATION

The update applies to both Audex-2B and Audex-30B-A3B. We retain the codec-sequence and decoded-audio notation $\mathbf{a}_{c,i}$ and $x_{c,i}$ from Section 4.1, and let $T_{c,i}$ count valid generated tokens.

Codec validity. Before judging, we truncate each sequence at the audio-generation end marker and check that it contains at least one complete frame and that every codec ID lies within its corresponding residual-vector-quantization (RVQ) codebook. A valid candidate receives margin $+1$ against an invalid candidate, whose margin is -1; two invalid candidates receive margin 0. These pairs bypass the judge but remain in the reward sum and comparison count. They can be represented in Equation (5) by unit mass on the valid candidate's win, or on TIE when both are invalid. These assigned distributions are not judge predictions.

Advantages and group selection. Using Equation (8), we compute

$$\widehat{A}_{c,i} = \frac{r_{c,i} - \mu_c}{\max(\sigma_c, \sigma_{\text{floor}})}. \tag{9}$$

The floor σ_{floor} limits amplification of small reward differences. Audex-2B discards entire groups with insufficient reward variation and samples replacements within a bounded budget; Audex-30B-A3B does not use this filtering or resampling. Normalization never pools candidates from different captions, and rewards and advantages remain fixed during the update. Appendix D.2 gives the model-specific settings.

Guided likelihoods and gradient paths. We evaluate Equation (3) on the same sampled tokens and prefixes for the current, old, and reference policies. All use the rollout guidance scale, temperature, and codebook-phase mask, with full softmax over the codec-valid support and no top-k or top-p truncation. For the GSPO gradient, we use

$$\widetilde{\mathbf{z}}^{\text{cfg}}_{\theta,t} = \gamma \mathbf{z}^{c}_{\theta,t} + (1-\gamma)\,\text{sg}\left(\mathbf{z}^{\emptyset}_{\theta,t}\right), \tag{10}$$

where sg is the identity in the forward pass with zero derivative. Applying the same masked softmax to $\widetilde{\mathbf{z}}^{\text{cfg}}_{\theta,t}/\tau$ preserves the probabilities in Equation (3) while stopping reward gradients through the current null computation.

Sequence-level updates. Let $\mathcal{B}$ be a batch of retained caption groups, $B = |\mathcal{B}|$, and $h_{c,i,t} = (c, \mathbf{a}_{c,i,<t})$. The old guided policy $\pi^{\text{cfg}}_{\theta_{\text{old}}}$ generates the rollout batch and is held fixed during its update. The length-normalized sequence ratio and GSPO loss (Zheng et al., 2025) are

$$\rho_{c,i}(\theta) = \exp\left[\frac{1}{T_{c,i}}\sum_{t=1}^{T_{c,i}} \log \frac{\pi^{\text{cfg}}_{\theta}(a_{c,i,t} \mid h_{c,i,t})}{\pi^{\text{cfg}}_{\theta_{\text{old}}}(a_{c,i,t} \mid h_{c,i,t})}\right], \tag{11}$$

$$\mathcal{L}_{\text{GSPO}}(\theta) = -\frac{1}{BG}\sum_{c\in\mathcal{B}}\sum_{i=1}^{G} \min\left\{\rho_{c,i}(\theta)\widehat{A}_{c,i},\ \text{clip}\big(\rho_{c,i}(\theta), 1-\epsilon_-, 1+\epsilon_+\big)\widehat{A}_{c,i}\right\}, \tag{12}$$

where ϵ_- and ϵ_+ are the lower and upper clipping widths. Each batch receives one optimizer step, followed by synchronization to the rollout engine and fresh sampling. Thus, current and old parameters match at loss evaluation, giving $\rho_{c,i}(\theta) \simeq 1$ up to numerical error and normally inactive clipping. This does not eliminate the policy gradient: old likelihoods are fixed while current likelihoods receive gradients. Each valid generated token shares the sequence advantage with $1/T_{c,i}$ length normalization.

C.2.2 GUIDED AND NULL-POLICY REGULARIZATION

Both penalties in Equation (4) use the frozen SFT initialization as their reference, distinct from the old policy refreshed for each rollout batch.

Sampled penalties. Using the same codec-valid support, the current null policy is

$$\pi_\theta(a_t \mid \emptyset, \mathbf{a}_{<t}) = \left[\text{softmax}\left(\mathbf{z}^{\emptyset}_{\theta,t}/\tau\right)\right]_{a_t}, \tag{13}$$

with the reference null policy defined analogously. On each token and prefix from the old guided rollout, we evaluate

$$\Delta^{\text{cfg}}_{c,i,t} = \log \frac{\pi^{\text{cfg}}_{\text{ref}}(a_{c,i,t} \mid c, \mathbf{a}_{c,i,<t})}{\pi^{\text{cfg}}_{\theta}(a_{c,i,t} \mid c, \mathbf{a}_{c,i,<t})},$$
$$\Delta^{\emptyset}_{c,i,t} = \log \frac{\pi_{\text{ref}}(a_{c,i,t} \mid \emptyset, \mathbf{a}_{c,i,<t})}{\pi_{\theta}(a_{c,i,t} \mid \emptyset, \mathbf{a}_{c,i,<t})}. \tag{14}$$

The null term replaces the caption with its length-matched null prompt. We use the nonnegative penalty k_3 with input and output caps:

$$k_3(u) = \exp(u) - 1 - u,$$
$$\kappa(u) = \min\left\{k_3\left(\text{clip}(u, -d_{\max}, d_{\max})\right), k_{\max}\right\}, \tag{15}$$

where $d_{\max} > 0$ and $k_{\max} > 0$. The regularizers are

$$\mathcal{L}^{s}_{\text{reg}}(\theta) = \frac{1}{|\mathcal{T}|} \sum_{(c,i,t)\in\mathcal{T}} \kappa\left(\Delta^{s}_{c,i,t}\right), \qquad s \in \{\text{cfg}, \emptyset\}, \tag{16}$$

where $\mathcal{T}$ contains valid generated-token positions, excluding caption and padding tokens. Unlike the sequence-normalized GSPO loss, these penalties are averaged directly over tokens. The null regularizer backpropagates through current null-policy log probabilities; it does not reuse detached null values from the GSPO computation.

Relationship to KL divergence. For normalized distributions p and q with common support, the uncapped function satisfies

$$\mathbb{E}_{a\sim p}\left[k_3\left(\log \frac{q(a)}{p(a)}\right)\right] = D_{\text{KL}}(p\|q). \tag{17}$$

At a fixed token history, taking $p = \pi^{\text{cfg}}_{\theta}$ and $q = \pi^{\text{cfg}}_{\text{ref}}$ gives a guided-policy KL interpretation when actions are sampled from p. Although current and old guided policies match before the update, our penalties include numerical caps and, for Audex-2B, reward-dependent group selection. The null penalty additionally uses guided rather than null-policy samples. We therefore describe both terms as sampled, KL-inspired regularizers, not unbiased estimates of their respective policy KL divergences. Appendix D.2 reports their settings, and Appendix E.3 analyzes coefficient sensitivity.

C.3 Out-of-Domain Reward Transfer: ETTA

This section specifies the DiffusionNFT (Zheng et al., 2026) objective used in Section 4.2. ETTA (Lee et al., 2025) generates continuous latents with a diffusion transformer and decodes them through an audio autoencoder. Optimization acts on completed latent generations; Appendix D.3 gives the rollout and training settings.

C.3.1 Preference Weights and Forward Noising

Preference weights. For each caption, we generate G clean latent candidates, decode them to waveforms, and convert the audio to 16-kHz mono for judging. After computing the rewards in Equation (2) and retaining groups with sufficient reward variation, we use the statistics in Equation (8) to obtain

$$A^{\text{NFT}}_{c,i} = \text{clip}\left(\frac{r_{c,i} - \mu_c}{\sigma_c}, -A_{\max}, A_{\max}\right),$$
$$w_{c,i} = \frac{1}{2} + \frac{A^{\text{NFT}}_{c,i}}{2A_{\max}} \in [0, 1], \tag{18}$$

where $A_{\max} > 0$ bounds the standardized advantage. The $w_{c,i}$ are optimizer-specific preference weights, not a redefinition of the shared rewards $r_{c,i} \in [-1, 1]$, and remain fixed during the update.

Forward noising. For each generated clean latent $\mathbf{z}_0$, we independently draw $t \sim p_{\text{time}}$ on $(0, 1)$ and Gaussian noise $\boldsymbol{\epsilon}$, then construct

$$\mathbf{z}_t = (1 - t)\mathbf{z}_0 + t\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{19}$$

Here, p_{time} is the noise-time distribution after the configured timestep shift. We draw fresh noise–time pairs while holding generated latents fixed. No gradients pass through sampling, and no sampling trajectories or their likelihoods are required.

C.3.2 Negative-Aware Flow-Matching Objective

Guided predictions. Let $\mathbf{v}_\theta(\mathbf{z}_t, t, c)$ and $\mathbf{v}_\theta(\mathbf{z}_t, t, \varnothing)$ be the conditional and null-text velocity predictions. We use

$$\mathbf{v}_\theta^{\text{cfg}}(\mathbf{z}_t, t, c) = \mathbf{v}_\theta(\mathbf{z}_t, t, \varnothing) + \gamma_{\text{D}} \left[\mathbf{v}_\theta(\mathbf{z}_t, t, c) - \mathbf{v}_\theta(\mathbf{z}_t, t, \varnothing)\right] \tag{20}$$

with the same γ_{D} for sampling, training, and the frozen reference model.

Positive and negative branches. For $\beta = 1$ and one update per rollout batch, write $\mathbf{u}_\theta = \mathbf{v}_\theta^{\text{cfg}}(\mathbf{z}_t, t, c)$ and define

$$\mathbf{u}_{\text{old}} = \text{sg}(\mathbf{u}_\theta), \qquad \mathbf{u}_\theta^+ = \mathbf{u}_\theta, \qquad \mathbf{u}_\theta^- = 2\mathbf{u}_{\text{old}} - \mathbf{u}_\theta. \tag{21}$$

The branches have identical forward values before the update but opposite derivatives with respect to the current prediction. This single-update setting does not require a separate old-policy model. For a velocity prediction $\mathbf{u}$, the clean-latent estimate and adaptively normalized reconstruction loss are

$$\begin{aligned} \widehat{\mathbf{z}}_0(\mathbf{u}) &= \mathbf{z}_t - t\mathbf{u}, \\ \ell(\mathbf{u}) &= \frac{\text{MSE}\left(\widehat{\mathbf{z}}_0(\mathbf{u}), \mathbf{z}_0\right)}{\text{sg}\left(\max\left\{\text{MAE}\left(\widehat{\mathbf{z}}_0(\mathbf{u}), \mathbf{z}_0\right), \delta\right\}\right)}. \end{aligned} \tag{22}$$

MSE and MAE average squared and absolute errors over latent elements. Each branch has its own detached normalization factor, with $\delta > 0$ ensuring numerical stability.

Preference optimization and reference regularization. With $\mathbf{u}_{\text{ref}} = \mathbf{v}_{\text{ref}}^{\text{cfg}}(\mathbf{z}_t, t, c)$ from the frozen initial ETTA checkpoint, we minimize

$$\mathcal{L}_{\text{ETTA}}(\theta) = \mathbb{E}\Big[\alpha_{\text{NFT}}\left\{w_{c,i}\ell(\mathbf{u}_\theta^+) + (1 - w_{c,i})\ell(\mathbf{u}_\theta^-)\right\} + \lambda_{\text{ref}}\,\text{MSE}(\mathbf{u}_\theta, \mathbf{u}_{\text{ref}})\Big]. \tag{23}$$

The expectation covers retained candidates and fresh noise–time draws; $\alpha_{\text{NFT}} > 0$ scales only the NFT term. Before the update, the opposite branch derivatives multiply the reconstruction gradient by $\alpha_{\text{NFT}}(2w_{c,i} - 1) = \alpha_{\text{NFT}} A_{c,i}^{\text{NFT}} / A_{\max}$. Above-average candidates promote reconstruction, below-average candidates act in the opposite direction, and zero advantage yields no NFT gradient. The reference penalty constrains guided velocity at the same noised latent, time, and caption; it is not an evaluated KL divergence between generation distributions. Each batch receives one update before fresh rollouts are generated.

D Training Configurations

This appendix summarizes the training configurations for *DuetJudge* and reward-guided optimization of Audex and ETTA. We report data sampling, optimization hyperparameters, and computational settings.

D.1 Reward Model Training

Table 11 summarizes the Stage-II data sampling coefficients and the training settings for both stages. For Holistic Preference, Fine-grained Alignment, and Perceptual Quality, the reported sample counts include both candidate orderings obtained by reversing each comparison pair, and are therefore twice the original numbers of comparison examples.

Table 11: Reward-model data sampling (upper) and training configurations (lower).

Training component	**Stage-II coefficient**	**Weighted training samples**
Holistic Preference	1.0	326,654
Fine-grained Alignment	1.0	130,304
Perceptual Quality	0.2	23,696
Audio Flamingo 3 understanding data	0.002	9,470
Music Flamingo understanding data	0.02	218,130
Music-theory question answering	0.5	7,436
General text instruction data	0.01	11,096

Setting	**Stage I**	**Stage II**
Optimizer	Adam	Adam
Adam (β_1, β_2)	$(0.9, 0.999)$	$(0.9, 0.999)$
Peak learning rate	2×10^{-5}	2×10^{-5}
Minimum learning rate (configured)	10^{-6}	–
Learning-rate schedule	Cosine	Constant after warmup
Warmup	2.5% of samples	50 steps
Weight decay	0.05	0.05
Gradient-norm clipping	1.0	1.0
Hidden / attention dropout	0.0/0.0	0.0/0.0
Training precision	BF16	BF16
Global batch size	128	128
Microbatch size per GPU	1	1
Maximum packed sequence length	131,072	131,072
Packing-buffer size	8,192	8,192
Number of GPUs	128	
Tensor / pipeline / context parallelism	4/1/1	
Data-parallel degree	32	
Gradient-accumulation steps	4	

After weighting, the non-tie-to-tie ratio was 10:1. Batch sizes and sequence lengths refer to packed multimodal sequences.

Implementation. Both stages use Megatron-LM (Shoeybi et al., 2019) with Transformer Engine, BF16 training, a distributed optimizer, and Energon sequence packing. The dataloader mixes datasets using nested epoch-based repetition coefficients and packs variable-length examples with a balanced greedy knapsack procedure. The batch sizes and sequence lengths in Table 11 therefore refer to packed multimodal sequences rather than individual comparisons. Training runs on 16 eight-GPU nodes with sequence parallelism and selective MLP activation recomputation.

Stage II is initialized from the Stage-I checkpoint at step 13,469, reusing its tokenizer. We fine-tune the full language model and audio projection adapter while keeping the audio encoder frozen, as specified in Appendix B.1. For pairwise examples, autoregressive cross-entropy is applied only to the assistant's final decision response, with all caption, instruction, and audio-input positions masked; capability-retention examples use their original textual response targets. The learning rate increases to 2×10^{-5} over 50 warmup steps and remains constant thereafter, with the remaining optimization settings reported in Table 11. Checkpoints are saved every 200 steps and evaluated offline. We use the checkpoint at step 400 as the frozen judge for downstream policy optimization.

D.2 Audex Policy Optimization

Rollout collection and batching. We implement CFG-aware GSPO for Audex-2B and Audex-30B-A3B using NeMo RL (nem, 2025), with FSDP2 policy training and colocated vLLM rollout generation. Table 12 summarizes their configurations alongside ETTA. The frozen judge compares all candidate pairs within each caption group in both orders. For Audex-2B, each update uses 16 retained caption groups with $G = 16$ candidates per caption, giving a global batch of 256 generated sequences. Groups with reward standard deviation at most 0.05 are discarded, and additional groups are collected until the training batch is filled, with a limit of ten generation batches. Audex-30B-A3B instead collects 48 captions with $G = 8$ candidates each, giving a global batch of 384, without dynamic group filtering or resampling. Both models use one optimizer update per collected training batch before synchronizing the updated parameters to the rollout engine.

Table 12: Policy-optimization configurations for Audex and ETTA. Rollout counts are reported before group filtering; algorithm-specific parameters are given in the corresponding text.

Setting	Audex-2B	Audex-30B-A3B	ETTA
Data and rollout collection			
Candidates per caption G	16	8	16
Captions per collection	16	48	12
Candidates per collection	256	384	192
Group-retention condition	$\sigma_c > 0.05$	No filtering	$\sigma_c \geq 0.05$
CFG scale (rollout / training)	3.0/3.0	3.0/3.0	3.5/3.5
Optimization			
Objective	GSPO	GSPO	DiffusionNFT
Optimizer		AdamW	
Learning rate	3×10^{-6}	3×10^{-6}	5×10^{-6}
Learning-rate schedule		Constant after warmup	–
Warmup steps		10	–
Gradient-norm clipping		1.0	
Updates per collected training batch		1	
Compute and checkpoints			
Policy precision (train / rollout)	FP32 / FP32	BF16 / BF16	FP32 / FP32
Policy / judge GPUs	16/16	48/16	6/2
Judge precision	BF16	BF16	BF16
Judge batch size per replica	16	16	16

Audex-2B resamples to obtain 256 retained candidates per update; Audex-30B-A3B uses all 384 candidates. ETTA's 192 candidates are counted before filtering and do not denote a fixed post-filtering optimization batch.

Guidance and codec constraints. Both Audex models use CFG scale $\gamma = 3.0$ and temperature $\tau = 1.0$ for rollout generation and policy-likelihood evaluation, without top-k or top-p truncation.

XCodec1 uses four interleaved codebooks at 50 frames per second. We cap generation at 2,000 codec tokens, corresponding to ten seconds, followed by the audio-generation end token; earlier termination is allowed at complete-frame boundaries. The same codebook-phase mask is applied during rollout generation and likelihood evaluation, so sampling uses the full softmax over the codec-valid support. The current model supplies the null branch, while the frozen SFT initialization supplies the reference policy.

Optimization and compute. Both models use a reward standard-deviation floor $\sigma_{\text{floor}} = 0.1$, GSPO clipping widths $(\epsilon_-, \epsilon_+) = (3 \times 10^{-4}, 4 \times 10^{-4})$, and fixed regularization coefficients $\lambda_{\text{cfg}} = 0.005$ and $\lambda_{\varnothing} = 0.1$. The regularizer input and output caps are $(d_{\max}, k_{\max}) = (20, 10)$. Training uses a microbatch size of one per GPU and a maximum total sequence length of 2,560. For Audex-30B-A3B, the ten-step linear warmup increases the learning rate from 0.1 times its configured value to the full value, followed by a constant schedule.

Audex-2B uses 16 policy GPUs and 16 judge GPUs across four nodes, with FP32 policy training and rollout generation. Audex-30B-A3B uses 48 policy GPUs and 16 judge GPUs across eight nodes, with BF16 policy computation and rollout generation while retaining FP32 master weights and optimizer states. Training uses tensor-parallel degree one for both models; rollout generation uses degrees one and two for Audex-2B and Audex-30B-A3B, respectively. The 30B run enables activation checkpointing and accumulates null-regularizer gradients through a separate backward pass before the same optimizer step to limit peak memory. Training and generation share the policy GPUs by alternating execution.

D.3 ETTA Policy Optimization

Rollouts and preference updates. We fine-tune ETTA's full generative backbone using Diffusion-NFT, keeping *DuetJudge* and the initial ETTA reference model frozen. Rollouts use 40-step DPM++ sampling (Lu et al., 2025) with CFG scale $\gamma_D = 3.5$; the same guidance scale is used for current and reference velocity predictions during training. We retain caption groups with reward standard deviation at least 0.05, clip standardized advantages to $[-5, 5]$, and map them to preference weights in $[0, 1]$. For each retained candidate, the update uses eight independently sampled noise–time pairs. Noise times are drawn uniformly before the configured timestep shift, and the generated clean latents remain fixed during optimization.

Optimization and compute. The NFT objective uses $\beta = 1$ and loss multiplier $\alpha_{\text{NFT}} = 5$, with reference-velocity regularization coefficient $\lambda_{\text{ref}} = 0.01$. We take one optimizer step per collected batch and then generate fresh candidates. The shared configuration fields are reported in Table 12. Training uses one eight-GPU node, with six training ranks and two BF16 judge replicas. Each rank collects two captions with 16 candidates each, yielding 12 caption groups and 192 candidates before filtering; the eight noise–time draws per retained candidate do not count as additional generated candidates. Reward-based early stopping is disabled, and checkpoints are evaluated offline.

E Additional Ablation Studies

This appendix complements Section 4.4 with the full candidate-order and reward-signal results, alternative comparison sets, and group-size and regularization sweeps. All experiments use Audex-2B; default training settings are reported in Appendix D.2.

E.1 Detailed Candidate-Order and Reward-Signal Ablations

We distinguish training-time swap augmentation, which adds label-consistent reversed pairs, from reward-time order averaging, which aligns candidate identities before combining the two presentation orders (Equation (5)). Table 13 reports all tested combinations of swap augmentation, order averaging, and probability-margin versus hard-decision rewards. The main-text summary in Table 5 retains the contrasts around the default configuration.

Swap augmentation and order averaging. With probability-margin rewards, swap augmentation improves all five automatic metrics whether or not order averaging is used. Order averaging in turn

Table 13: Full candidate-order and reward-signal ablations for Audex-2B RL. Rewards use normalized A/B/TIE probability margins or hard decisions; order averaging combines both presentation orders after aligning candidate identities. All RL comparisons are intra-group.

Reward signal	Order averaging	Audiobox Aesthetics ↑ (1–10)		CMI-RM ↑ (1–5)		MuQ ↑
		CE	PQ	Musicality	Alignment	
Audex-2B (*Base; no RL*)		6.852	7.257	0.651	1.006	0.209
RM training with swap augmentation						
Probability margin (default)	Yes	7.717	**8.123**	**1.303**	**1.435**	**0.296**
Probability margin	No	7.747	8.020	1.248	1.406	0.295
Hard decision	Yes	7.588	7.939	1.165	1.368	0.283
RM training without swap augmentation						
Probability margin	Yes	7.670	8.019	1.054	1.252	0.289
Probability margin	No	7.682	8.006	0.944	1.177	0.264
Hard decision	Yes	**7.818**	8.107	1.059	1.235	0.262

improves four of five metrics under either training setting, with a small decrease in Audiobox-CE. Its MuQ gain is larger without swap augmentation (0.264 to 0.289) than with it (0.295 to 0.296); for the swap-trained judge, it also raises Audiobox-PQ from 8.020 to 8.123 and CMI-RM Alignment from 1.406 to 1.435. These results support combining training-time augmentation and reward-time averaging rather than treating either as a substitute for the other.

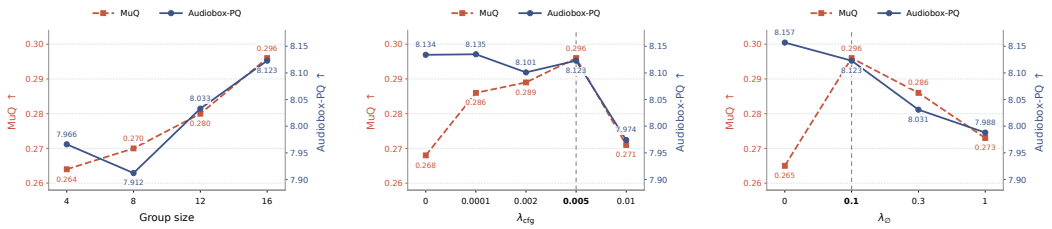
Probability margins versus hard decisions. With swap augmentation and order averaging fixed, the probability-margin rewards in Equation (2) improve all five automatic metrics over hard-decision rewards. Without swap augmentation, margins improve CMI-RM Alignment and MuQ but yield lower Audiobox scores and slightly lower CMI-RM Musicality. Across the variants in Table 13, the full recipe achieves the highest Audiobox-PQ, CMI-RM Musicality, CMI-RM Alignment, and MuQ, while hard decisions without swap augmentation achieve the highest Audiobox-CE. Together with the tie-supervision results in Table 5, these results support our default combination of tie-aware, swap-augmented training and order-averaged probability-margin rewards.

E.2 COMPARISON SET

Table 14 compares intra-group opponents, real-audio references, and their combination. The references are the source excerpts paired with noisy captions, rather than guaranteed preferred targets. Intra-group comparisons outperform reference-only comparisons on all five automatic metrics. Adding real references raises Audiobox-CE only from 7.717 to 7.721 while lowering the other four scores. These results favor comparing the current policy's own outputs: paired real recordings are not required for effective reward construction in this setting.

Table 14: Effect of the comparison set used to construct *DuetJudge* rewards for Audex-2B RL. Each generated sample is compared against other samples generated for the same caption (intra-group), the source real music excerpts paired with the noisy captions, or both.

Comparison set	Audiobox Aesthetics ↑ (1–10)		CMI-RM ↑ (1–5)		MuQ ↑
	CE	PQ	Musicality	Alignment	
Audex-2B (no RL)	6.852	7.257	0.651	1.006	0.209
Audex-2B with DuetJudge-guided RL					
Intra-group samples (default)	7.717	**8.123**	**1.303**	**1.435**	**0.296**
Real audio references	7.504	7.753	1.107	1.345	0.293
Intra-group + real references	**7.721**	8.051	1.173	1.380	0.283



(a) Rollout group size G. (b) Guided regularization λ_{cfg}. (c) Null regularization $\lambda_{\emptyset}$.

Figure 3: Sensitivity of Audex-2B RL to rollout group size and the regularization coefficients in Equation (4). Curves report MuQ and Audiobox-PQ; defaults are $G = 16$, $\lambda_{\text{cfg}} = 0.005$, and $\lambda_{\emptyset} = 0.1$.

E.3 Group Size and Policy Regularization

Group size. Figure 3a shows that increasing G from 4 to 16 raises MuQ monotonically from 0.264 to 0.296. Audiobox-PQ improves overall from 7.966 to 8.123 but dips at $G = 8$, so its response is not monotonic. The largest tested group, $G = 16$, yields the highest values on both metrics, at the cost of additional generations and pairwise judge evaluations.

Guided and null-policy regularization. Figures 3b and 3c vary the regularization coefficients in Equation (4). Raising λ_{cfg} from 0 to 0.005 improves MuQ from 0.268 to 0.296, while Audiobox-PQ changes from 8.134 to 8.123. Similarly, raising $\lambda_{\emptyset}$ from 0 to 0.1 improves MuQ from 0.265 to 0.296, with Audiobox-PQ decreasing from 8.157 to 8.123. The default coefficients achieve the highest MuQ in their respective sweeps, whereas larger coefficients lower both metrics. These results favor moderate regularization of both the guided policy and its unconditional baseline to improve caption alignment while largely preserving the production-quality score.

F Evaluation Protocols

F.1 Music Evaluators

The six baselines in Section 3.2 use scores from four pointwise music evaluators. Each evaluator scores one candidate at a time.

- **Audiobox Aesthetics** (Tjandra et al., 2025). We use *Content Enjoyment* (CE) and *Production Quality* (PQ) as separate scoring signals. CE estimates subjective listening appeal; PQ assesses technical quality, including clarity, fidelity, dynamics, and spatial presentation. Both predictors take only the waveform as input and are trained on human ratings on a 1–10 scale.
- **CMI-RM** (Ma et al., 2026). Given a clip and its conditioning inputs, CMI-RM predicts two scores: *Musicality*, which assesses musical and production quality, and *Alignment*, which measures adherence to the caption. We use the two scores separately.
- **MuQ-MuLan** (Zhu et al., 2025). This contrastively trained music–text embedding model scores the similarity between a clip and its caption. We report this score as *MuQ*.
- **TuneJury** (Kim et al., 2026). TuneJury uses pretrained audio and text representations with a learned scoring head to predict a single preference score for each caption–audio pair. Like CMI-RM, it is trained with pairwise preference supervision but scores candidates individually at inference.

F.2 Overall Preference Annotation

For the overall-preference evaluation in Section 3.2, we recruit 50 participants through *Prolific*. Each candidate pair is evaluated by at least two participants, who listen to both clips in full before responding.

Overall preference follows a two-stage rule: caption alignment takes priority, and sound quality is used only to break alignment ties. Participants first compare how well the two clips match the caption, ignoring differences in sound quality. If they choose A or B, that choice is recorded as the overall preference, and they proceed to the next pair. Only when they select "About the same" are they asked to compare sound quality. The second-stage response determines the final preference, with "About the same" recorded as a Tie. Figure 4 shows the second-stage interface.

We retain only candidate pairs for which all annotators agree on the final preference (*A*, *B*, or TIE).

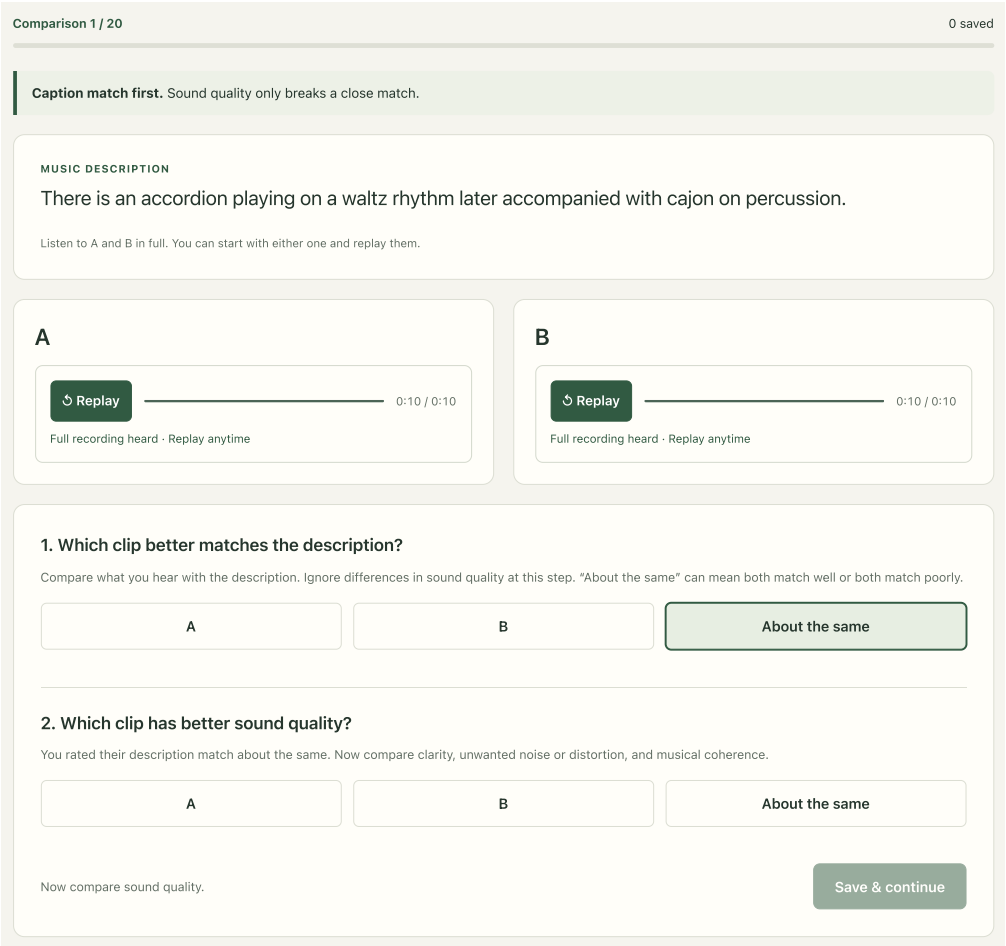


Figure 4: User Interface for Overall Preference Annotation. Sound-quality comparison shown only after participants select "About the same" for caption alignment.

F.3 AUTOMATIC EVALUATION

We report five automatic generation metrics: Audiobox Aesthetics CE and PQ, CMI-RM Musicality and Alignment, and MuQ-MuLan similarity (Appendix F.1). Each sample is scored independently, with the conditioning caption supplied when required.

F.4 HUMAN EVALUATION

We evaluate individual samples for overall quality (OVL) and caption relevance (REL), each on a five-point scale. Higher scores indicate better quality for OVL and a closer caption match for REL.

Overall quality (OVL). Participants rate the overall listening quality of each sample without seeing its caption. Figure 5 shows the evaluation interface.

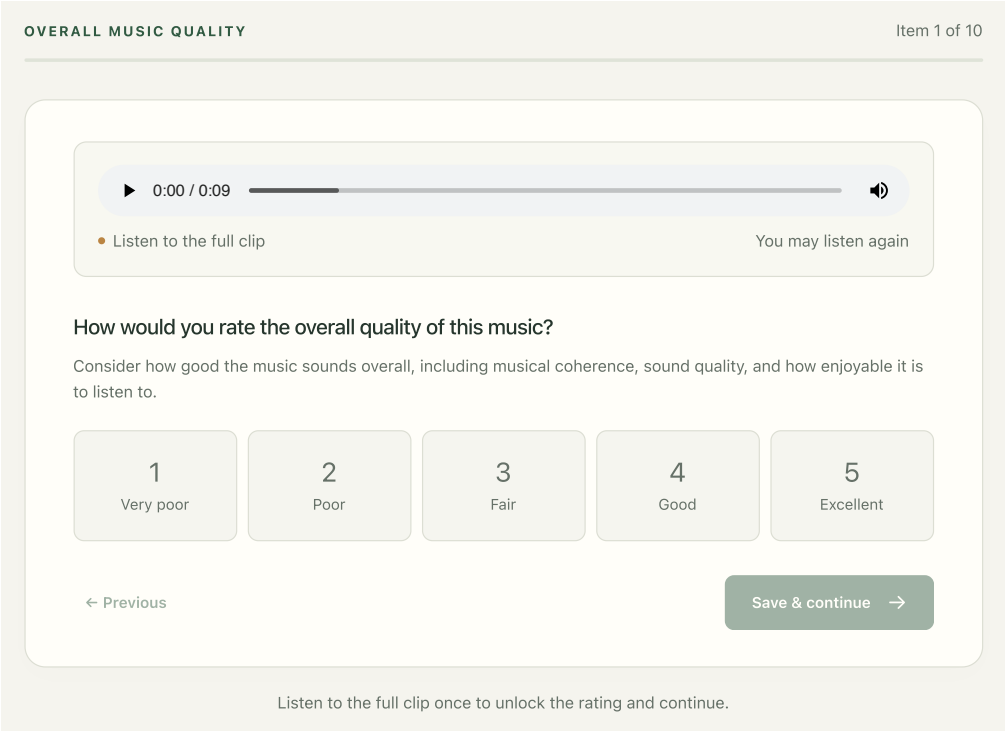


Figure 5: Human evaluation interface for overall quality (OVL).

Caption relevance (REL). Participants see the caption and rate how well the music matches the description. Figure 6 shows the evaluation interface.

Study setup. We recruit 50 participants through *Prolific* for this study. Recruitment is separate from that for the overall-preference annotation study (Appendix F.2). We evaluate 150 samples from each system, with each sample receiving 10 independent OVL ratings and 10 independent REL ratings. For each metric, we first average the ratings for each sample, then average the resulting sample-level scores for each system.

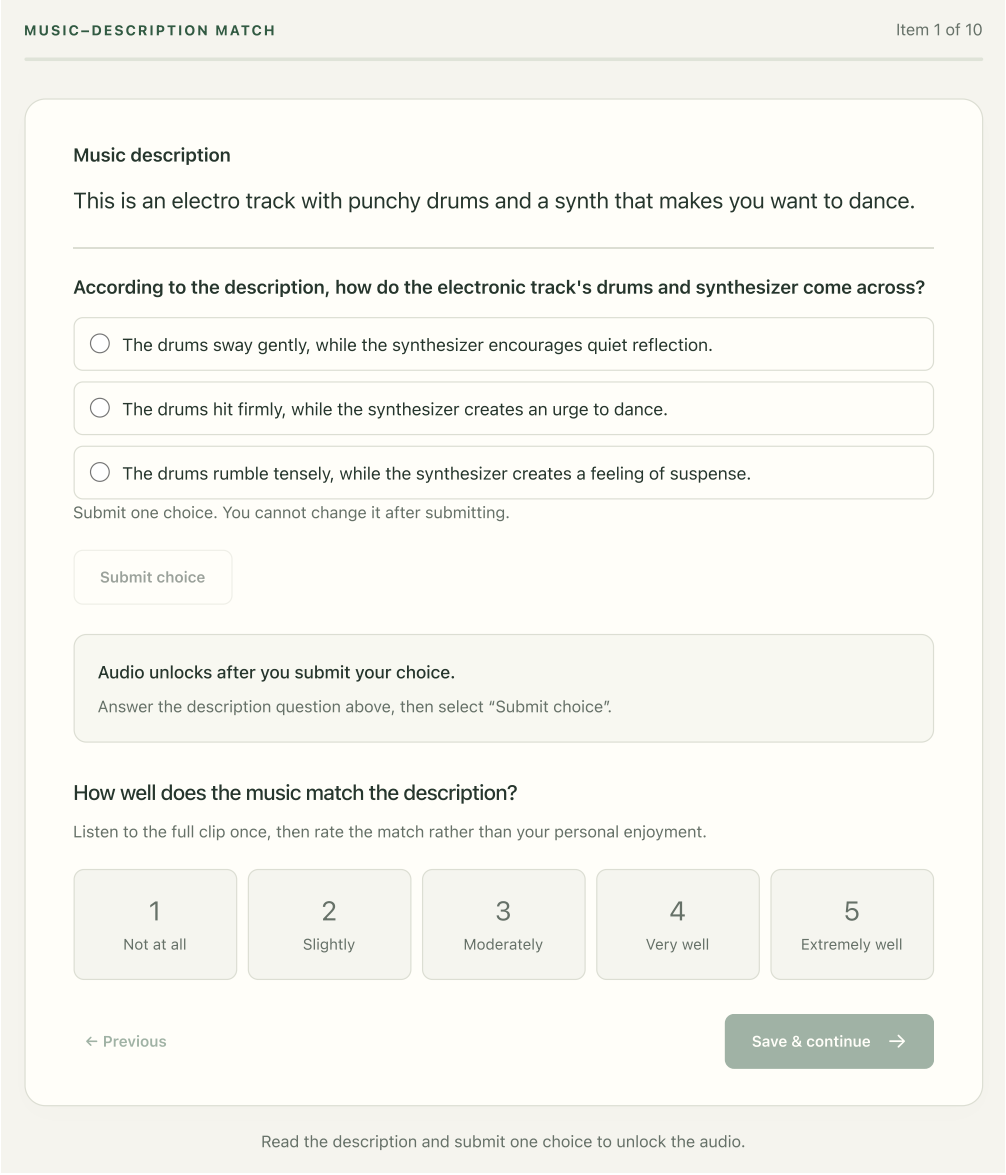


Figure 6: Human evaluation interface for caption relevance (REL).