SPHERE: Automatic Music Upmixing via Audio Language Model Post-Training with Spatial Heuristic Rewards

Zixun Guo[1,2]*, Calvin Murdock[1], Sanjeel Parekh[1], W. Owen Brimijoin[1], Simon Dixon[2], Joshua Reiss[2], Ishwarya Ananthabhotla[1]

[1]Meta Reality Labs, USA [2]Queen Mary University of London, UK

Project Demo Website

Abstract

In this paper, we study the task of automatic music upmixing, wherein a system predicts spatial mixing parameters from a multi-stem recording. Different from existing methods that rely on task-specific music encoders, we approach this task via audio language model (ALM) post-training, leveraging rich representations from existing ALMs, which encode both music semantics and mixing knowledge. Specifically, we propose a post-training recipe that first employs rejection sampling SFT, followed by reinforcement learning (RL) with verifiable rewards (RLVR) via GRPO. We propose SPHERE (Spatial Heuristic Rewards), a deterministic reward suite inspired by music mixing conventions, to guide our post-training. It consists of 6 perceptually-motivated sub-rewards and encourages the output mix to be centered, balanced and spacious. More broadly, our results suggest that expert domain knowledge can be encoded as verifiable rewards and distilled into language models, without task-specific architectures.

1 Introduction

Automatic music mixing is a well-studied task wherein a system receives isolated music stems and produces a music mix (De Man et al., 2017), inferring mixing parameters for each stem (gain, panning, etc.). In this paper, we address the task of automatic binaural upmixing, which extends the automatic music mixing task to the spatial domain (Begault, 1994). In particular, we additionally seek to predict spatial positioning parameters of each stem (i.e., azimuth, elevation) to create a musical presentation that appears three-dimensional and immersive in nature.

Existing deep learning based music mixing systems usually rely on task-specific encoders (e.g., CNN) to extract music representations to infer mixing parameters implicitly or explicitly (Steinmetz et al., 2021; Ramírez et al., 2022; Lee et al., 2024). We argue that these encoders are limited in their ability to encode musical semantics, such as the instrument grouping of stems, which plays a key role in a mix engineer's mixing process.

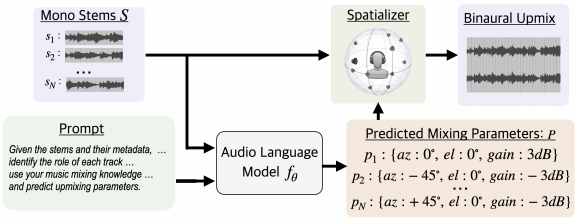

Figure 1: Automatic music upmixing via audio language modeling. Az. and El. stand for azimuth and elevation.

Meanwhile, recent Audio Language Models (ALMs) have shown superb audio understanding capabilities (Chu et al., 2023; Xu et al., 2025; Team, 2025; Ghosh et al., 2025). Since ALMs are usually also trained with internet-scale text, these models might have music mixing knowledge encoded (Doh et al., 2025). Hence, it is reasonable to believe when prompted with multitrack recording stems with mixing instructions, these models should be able to predict music mixing parameters, as shown in Figure 1. Indeed, our preliminary study shows that frontier ALMs understand music semantics in multi-track recordings and can produce decent mixing parameters zero-shot. This motivates us to tackle automatic music upmixing via ALMs, leveraging their rich pretrained representations. However, zero-shot ALMs do not produce mixes with consistent quality. We argue this can be addressed through supervised fine-tuning (SFT) and reinforcement learning (RL) post-training.

In this paper, we approach automatic binaural music upmixing via audio language model post-training, guided by SPHERE (Spatial Heuristic Rewards), a deterministic reward suite we propose that draws inspiration from music mixing conven-

*Work done at Meta.

Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing, pages 60827–60838
October 24-29, 2026 ©2026 Association for Computational Linguistics

tions, computed as follows. We first compute six perceptually-motivated raw metrics measuring centering, spread, and level balance of the mix. We then derive data-driven thresholds and apply shaping functions to convert raw metrics into continuous sub-rewards, which are then summed. SPHERE serves two roles: First, we utilize SPHERE to filter high-quality outputs from a teacher model and conduct SFT on a smaller student model. We then apply RL via GRPO (Shao et al., 2024) to further optimize the student against SPHERE, enabling it to explore better spatial parameters.

Our subjective listening test (A/B Preference) with 44 participants shows that high-reward mixes are preferred in 76.8% of trials, confirming SPHERE is a reliable proxy for human spatial audio preference. Moreover, after post-training, objective results on the MedleyDB dataset (Bittner et al., 2014) show that our model surpasses frontier ALMs by a large margin on SPHERE, demonstrating the effectiveness of the post-training pipeline. Last but not least, our ablation study shows that the combination of sub-rewards in SPHERE creates a reward landscape that cannot be trivially maximized: removing any pair leads to reward hacking. Only when all six components are present does RL training remain stable, achieving the highest total reward with the lowest variance.

In summary, our contributions are as follows:

1. To the best of our knowledge, we are the first to tackle automatic music upmixing via ALM post training, leveraging rich music understanding capabilities in pretrained ALMs.

2. We propose SPHERE (Spatial Heuristic Rewards), a reward function inspired by music mixing conventions, that encourages the final mix to be centered, balanced, and to have a wide spatial impression. Through a subjective listening test, we empirically show SPHERE aligns with human preference and prevents reward hacking during RL.

3. We show that our post-trained model outperforms frontier ALMs on our reward suite, which also serves as our evaluation metric.

4. More broadly, our results suggest that expert domain knowledge can be encoded as verifiable rewards and distilled into a language model through post-training. This offers a practical recipe for data-limited domains, potentially beyond music mixing.

2 Related Work

Existing deep learning approaches to automatic music mixing typically follow an encoder-decoder design, where an encoder extracts latent music representations, and a decoder predicts either interpretable mixing parameters (Steinmetz et al., 2021; Vanka et al., 2024; Lee et al., 2024) or fully-mixed audio directly (Ramírez et al., 2022; Moliner et al., 2026). For binaural upmixing, prior work has additionally relied on other modalities to learn mono or stereo to binaural conversion (Gao and Grauman, 2019; Lluís et al., 2022), or predict spatial audio directly from stereo inputs, inferring the spatial cues from stereo inputs implicitly (Grundhuber et al., 2024; Yang et al., 2022; Zang et al., 2024). In most cases, these methods have limited semantic understanding of multi-stem recordings or mixing conventions as the encoders are mostly CNN-based and trained from scratch, except Moliner et al. (2026), who use CLAP embeddings.

Meanwhile, recent audio language models (ALMs) have demonstrated strong audio and music understanding capabilities (Chu et al., 2023; Xu et al., 2025; Team, 2025). Some ALMs can process multiple audio files (e.g., multitrack music recordings) within a single prompt and produce descriptive text, making them suited for music mixing. However, general-purpose ALMs are not trained on spatial mixing data, resulting in inconsistent zero-shot performance. To improve task-specific performance, post-training via SFT then RL has proven effective for aligning LLM outputs with desired behaviors, via verifiable rewards (Shao et al., 2024; Hernandez et al., 2026) or human preferences (Cideron et al., 2024). Yet applying post-training to music upmixing requires a reward function that captures perceptual mixing quality and aligns with general listener preference. In this work, we aim to design such a reward function, drawing inspiration from music mixing conventions (Pestana and Reiss, 2014; Lopes et al., 2023).

3 Method

3.1 Task Formulation

The overall inference pipeline is shown in Figure 1. Let $\mathcal{S} = \{s_i\}_{i=1}^{N}$ denote the N mono stems of a multitrack recording, and T denote a text prompt that includes mixing instructions (De Man et al., 2019; Pestana and Reiss, 2014) as well as other descriptions of the stems (e.g., metadata, loudness in LUFS). We aim to learn a function $f_\theta(\mathcal{S}, T)$

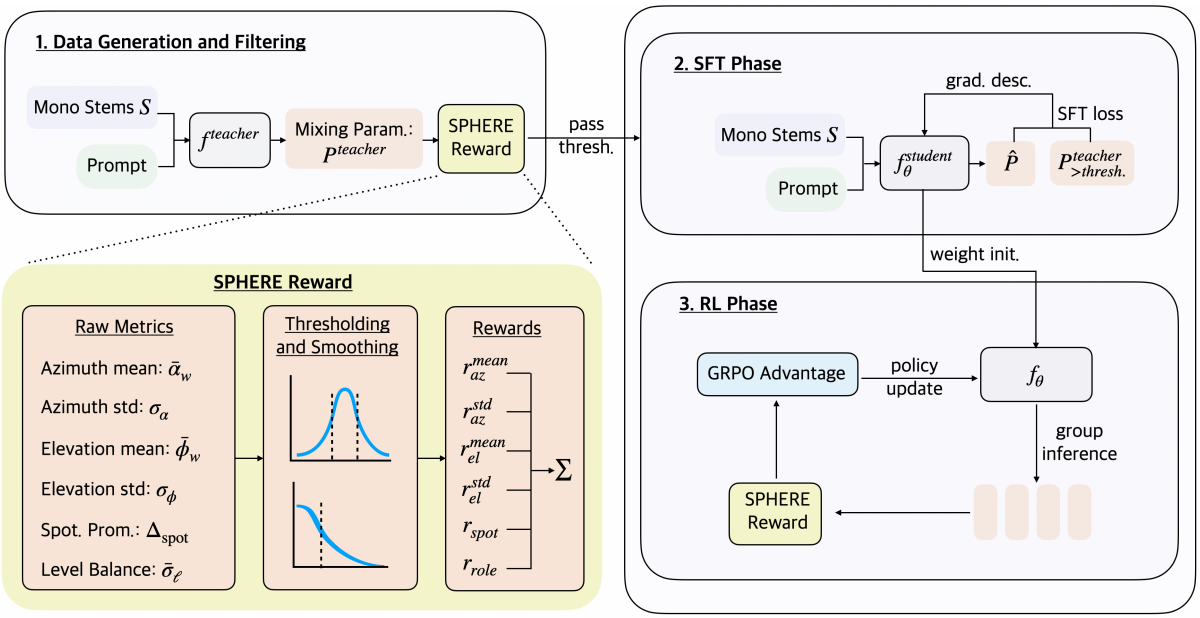

Figure 2: Overall post-training paradigm. 1. A teacher model generates spatial mixing parameters, which are filtered by R_{SPHERE}. 2. SFT finetunes a student on the filtered data. 3. RL further optimizes the student using R_{SPHERE}.

that maps $\mathcal{S}$ and T to a set of spatial parameters $P = \{p_i\}_{i=1}^{N}$, where $p_i = (\alpha_i, \phi_i, g_i)$ denotes the predicted azimuth, elevation (in degrees), and gain (in dB) of stem s_i. In practice, f_θ is implemented as an ALM that receives the mono stems as audio tokens interleaved with tokenized T, and produces P as JSON outputs. A binaural spatializer then applies the predicted gain g_i to each stem s_i and places it at position (α_i, ϕ_i) by convolving with a corresponding proprietary HRTF, producing a two-channel binaural mix for headphone playback.

3.2 Overall Post-Training Paradigm

The post-training paradigm is shown in Figure 2. To acquire paired data for SFT, a teacher model f_{teacher} generates candidate spatial parameters for each multitrack recording. To ensure data quality, we score each teacher output with R_{SPHERE} (to be introduced in Section 3.3) and we retain data points that exceed a quality threshold κ. A student model f_θ is then finetuned on the filtered high-quality data via SFT. Since SFT can only imitate the filtered teacher outputs, we further refine the student via RL against R_{SPHERE}, enabling it to discover spatial configurations beyond the teacher's distribution.

3.3 Reward Design

Next, we introduce SPHERE (Spatial Heuristic Rewards) for our post-training. The design principles governing our reward suite are as follows: first, it should rank spacious, balanced mixes above narrow, unbalanced ones, following music mixing conventions (Pestana and Reiss, 2014). Moreover, the reward function should reflect human perception of spatial audio aesthetics, with higher-reward data preferred by humans over lower-reward ones. Additionally, since it is used to calculate advantage scores during RL, it should provide a smooth gradient. In Section 3.3.1 we calculate six raw spatial metrics that quantify mix quality. We then convert these raw metrics into verifiable rewards via data-driven thresholding and gaussian and sigmoid smoothing (i.e., reward shaping) in Section 3.3.2.

3.3.1 Raw Spatial Metrics

Centering and Spread Metrics A spatially rich mix should be perceived as centered overall, and with individual stems spread across the panorama (Pestana and Reiss, 2014; Lopes et al., 2023). To reflect this, the centering and spread metrics are calculated as follows. For each stem i, let ℓ_i denote its original loudness (in dB LUFS) and recall that g_i, α_i are the predicted gain and azimuth respectively. We define the linear-scale weight $w_i = 10^{(\ell_i + g_i)/10}$, reflecting effective loudness after the gain is applied. For each spatial axis, we measure centering via a gain-weighted mean and spread via an unweighted standard deviation:

$$\bar{\alpha}_w = \frac{\sum_i w_i \alpha_i}{\sum_i w_i} \quad \sigma_\alpha = \sqrt{\frac{1}{N} \sum_i (\alpha_i - \bar{\alpha})^2} \quad (1)$$

Category	Raw Metric	Desired	Smooth.	Symbol
Centering	Azimuth mean $\bar{\alpha}_w$	$\to 0°$	Gaussian	$r_{\text{az}}^{\text{mean}}$
	Elevation mean $\bar{\phi}_w$	$\to 0°$	Gaussian	$r_{\text{el}}^{\text{mean}}$
Spread	Azimuth std σ_α	large	Sigmoid	$r_{\text{az}}^{\text{std}}$
	Elevation std σ_ϕ	large	Sigmoid	$r_{\text{el}}^{\text{std}}$
Level bal.	Spotlight prom. Δ_{spot}	≈ 0 dB	Gaussian	r_{spot}
	Per-role std σ_ℓ	$\to 0$	Gaussian	r_{role}

Table 1: Summary of sub-rewards in R_{SPHERE}.

Elevation centering $\bar{\phi}_w$ and spread metrics σ_ϕ are calculated analogously. The mean is gain-weighted because louder stems dominate perceived centering, while the standard deviation is unweighted because spread should reflect all stem positions equally (Blauert, 1997). This asymmetry also prevents potential reward hacking: σ_α cannot be inflated by manipulating gains alone. A desired spatial mix should have both mean metrics near zero (centered image) and both standard deviation metrics large (wide spatial impression).

Level Balance Metrics A good mix should also maintain balanced loudness relationships across stems, and with the spotlight track (e.g., lead vocal) clearly audible (Pestana and Reiss, 2014; Lopes et al., 2023). To reflect this, we measure two properties: spotlight prominence (Δ_{spot}) and per-role level consistency ($\sigma_\ell^{(c)}$). Spotlight prominence is measured as the loudness difference between the spotlight stems and the combination of remaining stems, and per-role level consistency is measured as the loudness's standard deviation among stems sharing the same musical role c (e.g., guitar and piano accompaniment). Both metrics require knowing each stem's role, which is not always available. Hence, we task the ALM with assigning each stem a role c, one of spotlight, foundation, percussion, accompaniment, or atmosphere, as part of its output. A well-balanced mix should have a near-zero spotlight prominence, indicating the spotlight tracks are clearly audible, and low per-role level deviation. The raw metrics, along with their corresponding category and desired direction are summarized in Table 1.

3.3.2 Data-driven Thresholds and Reward Shaping

Given the raw metrics and the desired directions in Table 1, we further use a function r to convert each raw metric into a bounded sub-reward $\in (0, 1]$, parametrized by cutoff thresholds τ that distinguish good outputs from poor ones. To avoid hand-selecting thresholds for each metric, which is hard to tune reliably, we propose to derive data-driven thresholds from the teacher model's output distribution. More specifically, for each metric we compute the p-th percentile boundary in the desired direction in Table 1. For instance, for centering metrics, whose desired raw values are near zero, the threshold τ can be derived as the radius from zero covering the top $p\%$ of teacher data points. For spread metrics, higher values indicate wider spatial spread. Hence, τ is the floor that excludes the narrowest $(1-p)\%$ of the teacher distribution.

After obtaining τ, the simplest solution is to apply a step function at each threshold ($r = 1$ when the metric satisfies the desired direction in Table 1, $r = 0$ otherwise). Although sufficient for SFT filtering, a step function provides sparse gradients for RL, as a sample just below the threshold receives the same zero reward as one far below. We instead replace the step function with a continuous relaxation, with the data-driven thresholds τ as the half-reward point: $r(\tau) = 0.5$ for each metric. Specifically, we apply a Gaussian for closer-to-target metrics and a Sigmoid for higher-is-better metrics in Table 1:

$$r_k = e^{-\frac{k^2}{2\sigma^2}}, \quad k \in \{\bar{\alpha}_w,\, \bar{\phi}_w,\, \Delta_{\text{spot}},\, \sigma_\ell^{(c)}\} \quad (2)$$

$$r_k = \frac{1}{1 + e^{-(k-\mu)/\gamma}}, \quad k \in \{\sigma_\alpha,\, \sigma_\phi\} \quad (3)$$

Here, the half-reward constraint $r_k(\tau) = 0.5$ determines the parameters for the reward functions: $\sigma = \tau/\sqrt{2\ln 2}$ for the Gaussian; $\mu = \tau$ and $\gamma = \tau/(3\ln 2)$ for the Sigmoid. Applying these shaping functions yields one sub-reward per metric, besides per-role level balance, since one recording can include stems spanning multiple roles, each with its own $\sigma_\ell^{(c)}$. We aggregate these sub-rewards via geometric mean over the K active roles, for those that contain more than one stem: $r_{\sigma_\ell} = \left(\prod_{c=1}^{K} r_{\sigma_\ell^{(c)}}\right)^{1/K}$. Here, the geometric mean acts as a soft "and": if any single role has high variance, r_{σ_ℓ} drops sharply, preventing the RL agent from sacrificing one role to optimize others. The total SPHERE reward sums all six sub-rewards, for $k \in \{\bar{\alpha}_w,\, \bar{\phi}_w,\, \Delta_{\text{spot}},\, \sigma_\ell^{(c)},\, \sigma_\alpha,\, \sigma_\phi\}$:

$$R_{\text{SPHERE}} = \sum_k r_k \quad (4)$$

For readability, Table 1 assigns a named symbol to each sub-reward, which we use in subsequent

sections. Notably, sub-reward pairs in SPHERE form a non-linear reward landscape, which makes reward optimization non-trivial. For instance, placing all stems at the center maximizes centering rewards but minimizes spread rewards, and vice versa. We later empirically verify that removing any sub-reward pair leads to reward hacking in Section 5.3, validating the reward design.

4 Experiments

4.1 Dataset

We use the MedleyDB multitrack dataset (Bittner et al., 2014) for our experiments, which contains 193 full-length songs spanning diverse genres, each provided with a stereo mix reference, isolated mono stems, and a metadata file. We segment each song's mono stems into aligned, non-overlapping 30-second windows, yielding 1650 unique data points. Due to context limitations, we allow a maximum of 29 mono stems (30 seconds each) per data point, resulting in input sequences of up to 20k audio tokens.

Following Section 3.1 and 3.2, we use Gemini 2.5 Pro as the teacher to generate candidate spatial configurations and use Qwen2.5-Omni (3B) as the student model. The prompt includes segmented mono stems and metadata, the stereo reference mix for additional reference, as well as a detailed mixing instruction. The full prompt is provided in Appendix A. For each segment, we run five independent inferences to obtain diverse spatial configurations, producing 8243 data points in total (7 failures). 23.5% of these data points are excluded, which include more than 29 stems, exceeding our student model's context budget. We then perform a song-level train-test split on the remaining 6304 data points to prevent timbre leakage across segments of the same song, yielding 15 test songs (618 data points) and 139 train songs (5686 data points). The percentile parameter p for data filtering thresholds in Section 3.3.2 is set to 0.8. Once the thresholds are defined, we retain only training data points where all six sub-rewards in Table 1 satisfy $r_k \geq \kappa$ and we set $\kappa = 0.5$. This produces 1408 high-quality data points for the student.

4.2 Training Setup

During the SFT phase, we finetune the student model on the filtered data via LoRA (rank 8, α=16) for 20 epochs, distributed across 16 H100 GPUs. The audio encoder is kept trainable. We use a learning rate of 2×10^{-4} with a cosine schedule and 5% linear warmup. Since each input can contain up to 29 mono stems, we set the max sequence length to 32768, the maximum supported by the student. Starting from an SFT checkpoint, we further improve the student with GRPO (Shao et al., 2024) for up to 800 steps. Due to the long input sequences, each RL step requires generating and scoring multiple full-length outputs, and an 800-step run takes approximately 24 hours on 8 H100 GPUs.

We employ an asynchronous training infrastructure that decouples policy optimization from generation: a training engine (6 H100s) updates the model weights while a separate inference engine (2 H100s) generates new spatial configurations asynchronously, with importance sampling correction to account for the off-policy lag between the two. For the training engine, we apply LoRA (rank 8, α=16) to all attention and MLP projections while freezing the audio encoder. We use PPO-style clipping with ϵ=0.2, and no KL penalty (Schulman et al., 2017). For the generation engine, at each step, a mini-batch of 6 input segments is sampled and the student generates G=4 spatial configurations per segment for GRPO, using temperature 1.0 and top-p=0.95.

4.3 Baselines and Ablations

Due to modality compatibility and sequence length constraints (details in Section 6), we only evaluate: 3 proprietary models (Gemini 2.5 Pro, 2.5 Flash and 2.0 Flash), and 5 open-source models (Qwen2.5-Omni-3B, 7B, and Qwen3-Omni Instruct, Captioner, and Thinking). All baselines are evaluated zero-shot on the same 618 test data points with identical prompts.

To study the effect of data quality versus quantity during SFT, we vary the filtering threshold $\kappa \in \{0.2, 0.3, 0.4, 0.5 \text{ (ours)}\}$ where a data point is accepted if all six sub-rewards satisfy $r_k \geq \kappa$. This yields 4074, 3152, 2283, and 1408 filtered data points respectively. To validate our reward design during RL, we conduct ablations along two axes. First, we replace the continuous reward shaping (Section 3.3.2) with a binary step function to verify whether continuous rewards provide a more effective learning signal for RL. Second, we disable pairs of sub-rewards corresponding to each category in Table 1 (i.e., spatial centering, spread and level balance) and analyze the training dynamics.

4.4 Evaluation

Subjective Evaluation To validate that SPHERE aligns with human preference, we conduct an AB listening test on teacher-generated mixes. We select 10 segments from the test set spanning diverse genres; for each, the teacher is prompted multiple times, from which we select a high- and low-reward mix with R_{SPHERE} gap of at least 1.0. We then render these 20 mixes using the spatializer described in Section 3.1 and loudness-normalized to -23 LUFS for fair comparison. In each trial, participants are presented with the two mixes in randomized order and indicate which spatial mix they prefer. Each trial takes 26 minutes on average.

We recruit 89 participants and apply a three-stage screening protocol to ensure reliable responses: headphone verification (Milne et al., 2021), gap detection (Musiek et al., 2005), and a spatial perception check (distinguish binaural from mono audio, identify which mix is more centered, and has better level balance). Of 89 participants, 44 passed all three stages (10 trials each, 440 total). The primary causes of exclusion were headphone verification failure (24%) and gap detection (21%).

We analyze the results with three statistical tests: (1) a one-sided binomial test on the overall preference rate across pooled trials, (2) a one-sided Wilcoxon signed-rank test on per-listener preference rates and (3) a paired Wilcoxon signed-rank test comparing R_{SPHERE} scores of preferred versus non-preferred mixes.

After post-training, we have additionally conducted a listening test that compares 4 systems: student, teacher, and our post-trained models (SFT, SFT+RL). We randomly sample 10 data points from our test set and render binaural upmixes from these 4 systems. We then recruit 10 expert listeners with music or spatial audio backgrounds following the same filtering protocol as above, and conduct a reference-free multi-stimulus preference test. Specifically, each participant is presented with the same track, but rendered from these 4 systems in random order. They are then asked to rate their preferences on a continuous 0-10 scale, with 0 and 10 indicating least and most preferred respectively.

Objective Evaluation We evaluate all models on the 618 held-out test data points, reporting R_{SPHERE}, all its sub-rewards and the format pass rate (whether the model produces valid JSON). To ensure our results reflect stable and representative performance, we report averaged SFT results over the last 100 training steps, where performance has converged. For RL, since the exploration is stochastic, we run each configuration 3 times independently and report the mean and standard deviation of the highest reward across runs. We additionally plot all sub-rewards against training steps to analyze convergence and potential reward hacking.

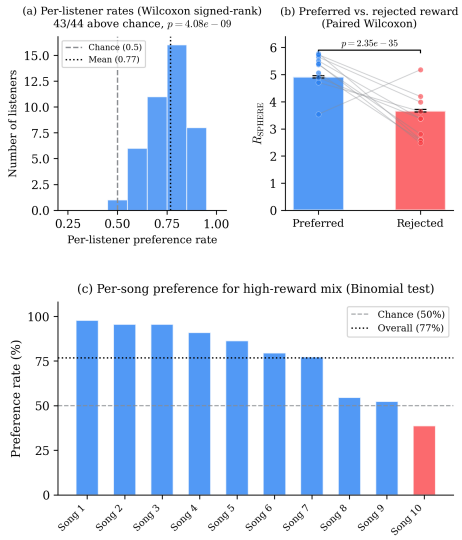

Figure 3: AB Listening test results (N=44 valid participants, 10 trials each). **(b)** Mean R_{SPHERE} of listener-preferred vs. listener-rejected mixes; gray lines connect paired scores for each song (10 pairs).

5 Results

5.1 Reward-Human Preference Alignment

We first verify whether SPHERE aligns with human preference and Figure 3 summarizes the results. Overall, the binomial test result (Figure 3 c) shows that listeners preferred the high-reward mix in 338 out of 440 trials (76.8%), which is significantly above chance, with $p = 6.65 \times 10^{-31}$. Moreover, the one-sided Wilcoxon signed-rank test confirms that individual preference rates are significantly above chance ($p = 4.08 \times 10^{-9}$, $r = 0.79$), with 43 out of 44 listeners individually preferring the high-reward mix (mean preference rate: 0.77 ± 0.11). Furthermore, in the paired Wilcoxon signed-rank test, mixes preferred by listeners had significantly higher R_{SPHERE} than the not-preferred mixes (4.92 ± 0.88 vs. 3.67 ± 1.01, $p = 2.35 \times 10^{-35}$). These results confirm that SPHERE is a meaningful proxy for human spatial audio preference, justifying the use of SPHERE as both the data-quality filter and the RL objective.

5.2 Post-Training Effectiveness

Table 2 summarizes the objective evaluation results. Among zero-shot baselines, Gemini 2.5 Pro achieves the highest $R_{\text{SPHERE}} = 4.548$, followed by Qwen3-Omni-Instruct at 4.180; Qwen2 models trail behind. We observe two notable failure patterns. First, Qwen2.5-Omni-3B (our student) scores the highest centering ($r_{\text{az}}^{\text{mean}} = 0.918$) but the lowest spread ($r_{\text{az}}^{\text{std}} = 0.177$), indicating that the model defaults to placing all sources at the center, achieving high centering at the expense of spatial diversity. Second, Qwen3-Omni-Thinking achieves a competitive $R_{\text{SPHERE}} = 4.278$ on valid outputs, but only 41.7% of its responses pass formatting: since our input contains multiple audio stems that can occupy over 20k of the 32k context window, the reasoning consumes the remaining budget before the model produces a spatial configuration. These two failure modes (i.e., collapsed spatial behavior and context-length exhaustion) illustrate the difficulty of zero-shot binauralization.

After SFT on reward-filtered teacher data, the same 3B student (SFT with $\kappa = 0.5$) reaches 4.657, even surpassing the teacher. RL further improves the total reward to 4.981, the highest score across all models. The full pipeline yields a gain of +1.407 over the untrained student, and even surpasses the much larger teacher model by a large margin, demonstrating that post-training with SPHERE is effective even on a small open-source model.

We next study per-metric changes to understand how SFT and RL individually improve the model. During SFT, the student first learns to produce valid outputs, with format pass rate improving from 80.1% to 98.8%. SFT also corrects the student's degenerate behavior (placing all sources at the center): centering metrics $r_{\text{az}}^{\text{mean}}$ and $r_{\text{el}}^{\text{mean}}$ decrease from 0.918 to 0.754 and 0.956 to 0.798 respectively. Meanwhile, spread metrics $r_{\text{az}}^{\text{std}}$ and $r_{\text{el}}^{\text{std}}$ increase dramatically, from 0.177 to 0.682 and 0.182 to 0.825 respectively. Moreover, level balance r_{role} also improves from 0.539 to 0.824.

RL further refines the student by targeting the weaker metrics after SFT by exploring the output space. The largest gains are in spotlight prominence r_{spot}, which improves from 0.774 to 0.879, and level balance r_{role} improves from 0.824 to 0.903. SFT left these two sub-rewards the most room for improvements. After RL, spread metrics further improve, from 0.682 to 0.759 for $r_{\text{az}}^{\text{std}}$, and 0.825 to 0.890 for $r_{\text{el}}^{\text{std}}$. These are at the cost of a slight decrease in elevation centering: $r_{\text{el}}^{\text{mean}}$ drops from 0.798 to 0.734. RL also resolves the remaining formatting failures, resulting in 100% correct format rate. In summary, SFT improves the student by imitating teacher behavior, but remains bounded by the quality of the teacher data. RL further improves this by reinforcing the model's own high-reward outputs and shifts the output distribution towards higher-quality spatial configurations.

5.3 Ablation Studies

Results of our ablation studies are shown in Table 2 and Figure 4. We report 2 findings: (1) data quality is more important than quantity during SFT and (2) removing sub-reward pairs can lead to reward hacking during RL. For the SFT data filtering ablation (Table 2), stricter filtering ($\kappa = 0.5$) retains only 1408 data points, yet achieves the highest R_{SPHERE} of 4.657, outperforming $\kappa = 0.2$, which uses nearly 3 times more data, yet reaches only 4.517. More broadly, R_{SPHERE} increases monotonically with κ: from 4.517 at $\kappa = 0.2$, to 4.605 at $\kappa = 0.3$, 4.609 at $\kappa = 0.4$, and 4.657 at $\kappa = 0.5$. This confirms that data quality matters more than quantity for SFT.

Next, we ablate individual pairs of RL sub-rewards according to Table 1. Figure 4 plots the sub-rewards over training steps for each configuration. Notably, removing any pair of sub-rewards leads to reward hacking, with severity depending on the removed components. The most severe case is removing the spatial spread sub-rewards ($r_{\text{az}}^{\text{std}}, r_{\text{el}}^{\text{std}}$). Without the incentive to spread sources apart, the model (labeled in green) learns to place all stems at the center to maximize centering, but spread metrics collapse. This leads to the lowest total reward of 4.388 among all RL configurations. Removing the centering sub-rewards ($r_{\text{az}}^{\text{mean}}, r_{\text{el}}^{\text{mean}}$) produces the mirror effect: the model (labeled in yellow) pushes stems apart to maximize spread, with centering dropping to 0.681 while spread inflates to 0.787. In both cases, the model exploits the remaining sub-reward at the expense of its removed counterparts. Removing the level balance sub-rewards ($r_{\text{spot}}, r_{\text{role}}$) does not result in severe reward hacking, as shown in the model labeled in orange, but spotlight prominence and per-role level balance both degrade, while the remaining sub-rewards stay stable. Reward hacking is possible in this case too, where the model could place a few loudest stems in the middle, while scattering quieter stems to inflate spread sub-rewards.

Model	$r_{\text{az}}^{\text{mean}}$	$r_{\text{az}}^{\text{std}}$	$r_{\text{el}}^{\text{mean}}$	$r_{\text{el}}^{\text{std}}$	r_{spot}	r_{role}	**Fmt%**	**Total**
Baselines								
Gemini 2.5 Pro (Teacher)	0.797	0.705	0.729	0.849	0.744	0.724	100.0	4.548
Gemini 2.5 Flash†	0.741	0.609	0.673	0.708	0.653	0.577	98.3	3.961
Gemini 2.0 Flash†	0.589	0.645	0.524	0.855	0.907	0.513	95.6	4.033
Qwen2.5-Omni-3B (Student)	0.918	0.177	0.956	0.182	0.802	0.539	80.1	3.574
Qwen2.5-Omni-7B	0.855	0.270	0.900	0.309	0.810	0.485	94.2	3.630
Qwen3-Omni-30B-A3B-Instruct	0.743	0.580	0.679	0.731	0.784	0.664	98.9	4.180
Qwen3-Omni-30B-A3B-Captioner	0.623	0.549	0.611	0.749	0.664	0.437	98.4	3.632
Qwen3-Omni-30B-A3B-Thinking‡	0.677	0.602	0.795	0.652	0.775	0.778	41.7	4.278
SFT Phase (avg. over last 100 training steps)								
SFT (κ=0.2)	0.751	0.668	0.797	0.810	0.694	0.796	100.0	4.517±.060
SFT (κ=0.3)	0.781	0.656	0.800	0.815	0.731	0.822	99.8	4.605±.018
SFT (κ=0.4)	0.765	0.659	0.797	0.825	0.744	0.819	99.9	4.609±.052
SFT (κ=0.5, ours)	0.754	0.682	0.798	0.825	0.774	0.824	98.8	4.657±.032
RL Phase (each avg. over 3 separate runs)								
SFT + RL (w/o smoothing)	0.777	0.713	0.813	0.844	0.781	0.848	99.7	4.775±.070
SFT + RL (w/o $r_{\text{az}}^{\text{std}}, r_{\text{el}}^{\text{std}}$)	0.947	0.275	0.955	0.401	0.863	0.948	100.0	4.388±.088
SFT + RL (w/o $r_{\text{az}}^{\text{mean}}, r_{\text{el}}^{\text{mean}}$)	0.681	0.787	0.733	0.896	0.823	0.862	99.9	4.782±.080
SFT + RL (w/o $r_{\text{spot}}, r_{\text{role}}$)	0.806	0.740	0.799	0.859	0.778	0.809	99.8	4.791±.057
SFT + RL (ours)	0.816	0.759	0.734	0.890	0.879	0.903	100.0	**4.981±.013**

Table 2: Objective evaluation results. Each r represents a sub-reward in R_{SPHERE} (higher is better). Fmt% is the percentage of outputs with valid formatting. κ is minimum per-metric threshold for SFT filtering. †Evaluated on N=573; 45 items with 29 or more stems were excluded due to API failure. ‡Low format pass rate: chain-of-thought reasoning exhausts the context window.

However, this would require the model to deviate significantly from the SFT distribution, and we argue that PPO clipping limits how far the policy can shift per step, potentially preventing dramatic reward hacking. Finally, when all 6 sub-rewards are present (model labeled in darker blue), R_{SPHERE} achieves the highest total with no metric collapsing, suggesting that the reward combination is resistant to hacking. Moreover, replacing continuous reward shaping with a binary step function (model labeled in lighter blue) reduces total reward from 4.981 to 4.775, with noticeably higher variance across runs in Table 2. This confirms that the Gaussian and sigmoid shaping introduced in Section 3.3.2 provides a more effective learning signal for RL.

5.4 Subjective Evaluation of Post-Trained Models

Model	Mean	Std
Qwen2.5-Omni-3B (Student)	3.24	2.20
Gemini 2.5 Pro (Teacher)	5.55	2.80
SFT (ours)	5.71	2.74
SFT+RL (ours)	5.73	2.85

Table 3: Listening test results that compare models before and after post-training (10 expert listeners, 10 tracks). Preference ratings on a continuous 0 to 10 scale, where 10 is most preferred.

We further compare our post-trained models with the student and teacher model via a multi-stimulus listening test. The results in Table 3 show that the student model is the least preferred. After SFT and RL post-training, the overall ratings are improved to 5.71 and 5.73, outperforming the teacher model, with a rating of 5.55. Interestingly, although the listening test result follows the same trend as the objective R_{SPHERE} in Table 2 (SFT+RL ranks highest, followed by SFT, the teacher, and the student), we are seeing smaller improvements from SFT to SFT+RL (5.71 to 5.73), compared to the SFT phase (3.24 to 5.71). This shows potential further directions when designing reward suites for post-training: each sub-reward might have a different perceptual weight (e.g., spotlight prominence might affect the overall preference more easily than centering metrics). We leave a systematic study of perceptually-weighted rewards to future work.

6 Limitations and Future Directions

While effective for automatic music binaural upmixing, our approach has two main limitations. First, it is constrained by modality compatibility, as most existing frontier models have limited support for the audio modality. Among major frontier model families as of 2026 (e.g., Claude, GPT, Gemini, DeepSeek, and Qwen), only Gemini, Qwen,

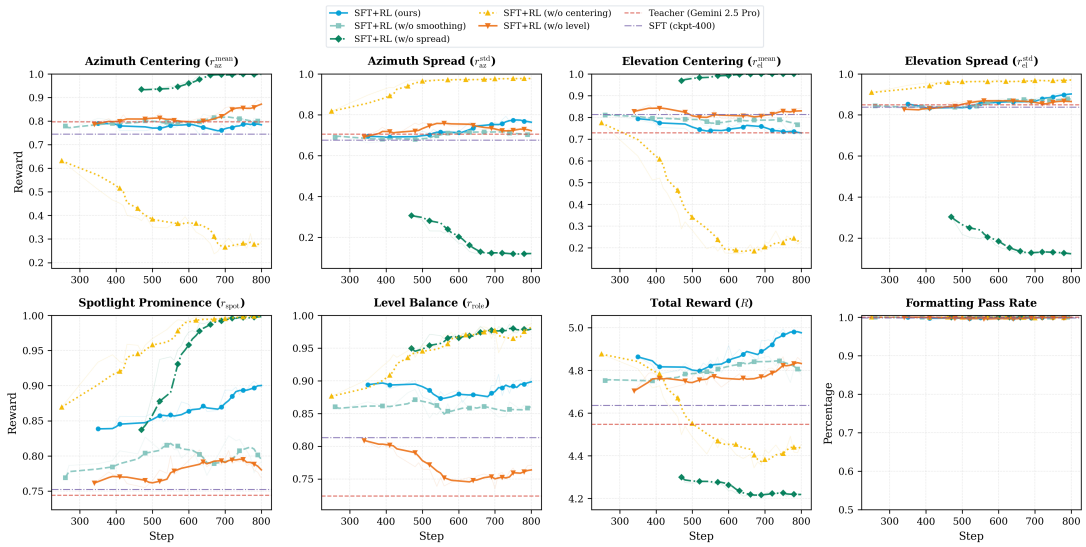

Figure 4: RL training dynamics. Removing individual sub-reward pairs leads to reward hacking on remaining metrics. RL with continuous relaxation outperforms the step-function baseline.

Gemma, and GPT-4o support audio input. Among these, GPT-4o and Gemma3N are primarily designed for speech understanding, while Gemma3N supports only a single audio input per prompt. Second, context length is a main bottleneck. The Qwen models tokenize audio at 25 tokens per second (Xu et al., 2025). A single 30-second stem thus consumes around 750 tokens. With prompt overheads, the 32k-token context window accommodates at most 29 30-second stems, yet real-world multitrack recordings often exceed this limit and contain full-length audio beyond 30 seconds. This caused the exclusion of 23.5% of our dataset in Section 4.1. This large context window also makes both SFT and RL training expensive. Future work could explore context-length scaling techniques or efficient tokenizers to accommodate multitrack recordings.

7 Conclusion

We presented an approach to automatic music binaural upmixing via ALM post-training, guided by SPHERE, a reward inspired by music mixing conventions. We empirically verify that SPHERE aligns with human preference and through post-training, our student model surpasses frontier ALMs. Our ablation studies show that SFT data quality is more important than quantity, and all 6 sub-rewards are necessary to prevent reward hacking during RL. More broadly, our work suggests that musical domain expertise can be distilled into reward functions to guide post-training, enabling optimization of ALM outputs with respect to common practices.

References

Durand R. Begault. 1994. *3-D Sound for Virtual Reality and Multimedia*. Academic Press, San Diego, CA.

Rachel M. Bittner, Justin Salamon, Mike Tierney, Matthias Mauch, Chris Cannam, and Juan Pablo Bello. 2014. Medleydb: A multitrack dataset for annotation-intensive MIR research. In *Proceedings of the 15th International Society for Music Information Retrieval Conference, ISMIR 2014, Taipei, Taiwan, October 27-31, 2014*, pages 155–160.

Jens Blauert. 1997. *Spatial Hearing: The Psychophysics of Human Sound Localization*, revised edition. MIT Press, Cambridge, MA.

Yunfei Chu, Jin Xu, Xiaohuan Zhou, Qian Yang, Shiliang Zhang, Zhijie Yan, Chang Zhou, and Jingren Zhou. 2023. Qwen-audio: Advancing universal audio understanding via unified large-scale audio-language models. *arXiv preprint arXiv:2311.07919*.

Geoffrey Cideron, Sertan Girgin, Mauro Verzetti, Damien Vincent, Matej Kastelic, Zalán Borsos, Brian McWilliams, Victor Ungureanu, Olivier Bachem, Olivier Pietquin, Matthieu Geist, Léonard Hussenot, Neil Zeghidour, and Andrea Agostinelli. 2024. Musicrl: Aligning music generation to human preferences. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pages 8968–8984. PMLR / OpenReview.net.

Brecht De Man, Joshua D. Reiss, and Ryan Stables. 2017. Ten years of automatic mixing. In *Proceedings of the 3rd Workshop on Intelligent Music Production*.

Brecht De Man, Ryan Stables, and Joshua D. Reiss. 2019. *Intelligent Music Production*. Focal Press, London, UK.

Seungheon Doh, Junghyun Koo, Marco A Martínez-Ramírez, Wei-Hsiang Liao, Juhan Nam, and Yuki Mitsufuji. 2025. Can large language models predict audio effects parameters from natural language? In *2025 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE.

Ruohan Gao and Kristen Grauman. 2019. 2.5d visual sound. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE.

Sreyan Ghosh, Arushi Goel, Lasha Koroshinadze, Sanggil Lee, Zhifeng Kong, Joao Felipe Santos, Ramani Duraiswami, Dinesh Manocha, Wei Ping, Mohammad Shoeybi, and Bryan Catanzaro. 2025. Music flamingo: Scaling music understanding in audio language models. *arXiv preprint arXiv*.

Philipp Grundhuber, Michael Lovedee-Turner, and Emanuël A. P. Habets. 2024. NBU: Neural binaural upmixing of stereo content. In *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, pages 404–411.

Jefferson Hernandez, Swarnadeep Saha, Chenxi Whitehouse, Sanjeel Parekh, Calvin Murdock, Yuliang Li, W Owen Brimijoin, Vamsi Krishna Ithapu, and Ishwarya Ananthabhotla. 2026. Text-to-stage: Spatial layouts from long-form narratives. *arXiv preprint arXiv:2603.17832*.

Sungho Lee, Marco A. Martínez Ramírez, Wei-Hsiang Liao, Stefan Uhlich, Giorgio Fabbro, Kyogu Lee, and Yuki Mitsufuji. 2024. Searching for music mixing graphs: A pruning approach. *CoRR*, abs/2406.01049.

Francesc Lluís, Vasileios Chatziioannou, and Alex Hofmann. 2022. Points2sound: from mono to binaural audio using 3d point cloud scenes. *EURASIP J. Audio Speech Music. Process.*, 2022(1):33.

Afonso Lopes, José Ricardo Barboza, and Gilberto Bernardes. 2023. Instrument position in immersive audio: An empirical review of award-winning practices. In *2023 Immersive and 3D Audio: from Architecture to Automotive (I3DA)*, pages 1–8.

Alice E. Milne, Roberta Bianco, Katarina C. Poole, Sijia Zhao, Andrew J. Oxenham, Alexander J. Billig, and Maria Chait. 2021. An online headphone screening test based on dichotic pitch. *Behavior Research Methods*, 53:1551–1562.

Eloi Moliner, Marco A. Martínez-Ramírez, Junghyun Koo, Wei-Hsiang Liao, Kin Wai Cheuk, Joan Serrà, Vesa Välimäki, and Yuki Mitsufuji. 2026. Automatic music mixing using a generative model of effect embeddings. In *ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 14582–14586.

Frank E. Musiek, Jennifer B. Shinn, Robert Jirsa, Doris-Eva Bamiou, Jane A. Baran, and Eda Zaidan. 2005. GIN (gaps-in-noise) test performance in subjects with confirmed central auditory nervous system involvement. *Ear and Hearing*, 26(6):608–618.

Pedro D. Pestana and Joshua D. Reiss. 2014. Intelligent audio production strategies informed by best practices. In *AES International Conference on Semantic Audio 2014, London, UK, January 27-29, 2014*. Audio Engineering Society.

Marco A. Martínez Ramírez, Wei-Hsiang Liao, Chihiro Nagashima, Giorgio Fabbro, Stefan Uhlich, and Yuki Mitsufuji. 2022. Automatic music mixing with deep learning and out-of-domain data. In *Proceedings of the 23rd International Society for Music Information Retrieval Conference, ISMIR 2022, Bengaluru, India, December 4-8, 2022*, pages 411–418.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300.

Christian J. Steinmetz, Jordi Pons, Santiago Pascual, and Joan Serrà. 2021. Automatic multitrack mixing with A differentiable mixing console of neural audio effects. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2021, Toronto, ON, Canada, June 6-11, 2021*, pages 71–75. IEEE.

Gemini Team. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *CoRR*, abs/2507.06261.

Soumya Sai Vanka, Christian J. Steinmetz, Jean-Baptiste Rolland, Joshua D. Reiss, and George Fazekas. 2024. Diff-mst: Differentiable mixing style transfer. In *Proceedings of the 25th International Society for Music Information Retrieval Conference, ISMIR 2024, San Francisco, California, USA and Online, November 10-14, 2024*, pages 563–570.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, Bin Zhang, Xiong Wang, Yunfei Chu, and Junyang Lin. 2025. Qwen2.5-omni technical report. *arXiv preprint arXiv:2503.20215*.

Haici Yang, Sanna Wager, Spencer Russell, Mike Luo, Minje Kim, and Wontak Kim. 2022. Upmixing via style transfer: A variational autoencoder for disentangling spatial images and musical content. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2022, Virtual and Singapore, 23-27 May 2022*, pages 426–430. IEEE.

Yongyi Zang, Yifan Wang, and Minglun Lee. 2024. Ambisonizer: Neural upmixing as spherical harmonics generation. *arXiv preprint arXiv:2405.13428*.

Appendix

A Prompt Template

The full prompt is shown in Figure 5. It outputs a per-stem JSON configuration specifying azimuth, elevation, gain, musical role, and a reasoning trace. The prompt structures gain prediction as a three-stage process: gain staging normalization, aesthetic balance, and compensation. Role assignment is performed by the model rather than provided as input, since metadata does not always indicate musical function. Spatial placement principles drawn from mixing conventions (Pestana and Reiss, 2014) are included directly in the prompt.

B SFT Training Dynamics

Figure 6 plots the reward dynamics on the test set throughout SFT for each filtering threshold κ. Three phases are visible across all configurations. In the first $\sim$100 steps, the model rapidly learns output formatting and corrects its degenerate zero-shot behavior: azimuth and elevation spread climb sharply from their collapsed baselines, while centering metrics decrease as the model begins distributing stems across the panorama rather than clustering them at center. Between steps $\sim$100-200 all sub-rewards quickly increase and slowly converge towards their final reward value after step $\sim$200.

The $\kappa = 0.5$ model achieves the highest total reward despite using the fewest training samples, confirming the data-quality-over-quantity finding reported in Section 5.3. The red dashed line (teacher performance) is surpassed by all SFT configurations on total reward, demonstrating that even basic reward filtering enables the student to exceed the teacher.

C Label-to-Role Assignment Analysis

As discussed in Section 3.3.1, the level balance sub-rewards rely on roles predicted by the ALM, raising the question of whether these assignments are reliable. Since no ground truth exists for the full role taxonomy, direct evaluation is difficult. However, the MedleyDB dataset provides two component labels (i.e., melody and bass), which we use for partial validation and analysis. Table 4 reports label-to-role assignment distribution, mapping MedleyDB's ground-truth labels to predicted track roles. Bass $\rightarrow$ Fnd. measures the percentage of bass-labeled stems assigned `foundation`. For stems labeled as melody, we report the percentage assigned `spotlight`, `accompanying_parts`, or other roles.

Table 4 shows that bass stems are assigned `foundation` most of the time by all ALMs, whereas melody-labeled stems are primarily assigned as `spotlight` or `accompanying_parts` by ALMs. This reveals a fundamental challenge in music semantics research: existing dataset annotations are static and song-level, whereas musical roles are inherently time-variant and context-dependent. For instance, a jazz piece: Oil, in the MedleyDB dataset has both the piano and guitar stems labeled as `melody`. Yet in a given segment the guitar solos while the piano comps. The ALM correctly assigns them `spotlight` and `accompanying_parts` respectively. In another classical piece (MusicDelta Beethoven), all instruments (flute, clarinet, oboe, trumpet, French horn) share the metadata label: melody, yet ALMs assign them to `others`, which is also semantically correct since they do not belong to any suitable roles we defined in Section 3.3.1.

Despite these challenges, Table 4 indicates our post-trained model assigns roles based on segment-level musical context rather than defaulting to metadata-driven shortcuts (e.g., always assigning `spotlight` to stems labeled `melody` in the prompt). Our analysis also reveals a fundamental challenge: text descriptions of music are usually static, whereas we empirically show that music is inherently context-dependent, and song-level description would lead to information bottlenecks. We believe this gap presents an opportunity for future work in multimodal music representation learning.

Model	Bass→Fnd.	Mel→Spot.	Mel→Acc.	Mel→Oth.
Baselines				
Gemini 2.5 Pro (Teacher)	97.9	30.7	53.1	16.2
Gemini 2.5 Flash	100.0	31.4	67.6	0.9
Gemini 2.0 Flash	94.2	12.1	54.0	33.9
Qwen2.5-Omni-3B (Student)	93.9	30.1	50.4	19.4
Qwen2.5-Omni-7B	93.9	29.6	24.1	46.3
Qwen3-Omni-Instruct	97.1	25.3	72.2	2.5
Qwen3-Omni-Captioner	99.7	33.2	41.3	25.5
Qwen3-Omni-Thinking	100.0	39.6	57.5	2.8
Post-trained				
SFT (κ=0.5, ours)	99.3	46.2	51.3	2.5
SFT + RL (ours)	99.7	26.9	59.5	13.6

Table 4: Label-to-Role assignment analysis using MedleyDB `component` labels and ALM predicted roles. MedleyDB's `component` label is a static song-level label whereas roles assigned by ALM are segment-level.

Full Prompt

```
system_prompt: You are an expert audio engineer and music producer.

user_prompt:
  1. Context & Inputs
  You are a Spatial Audio Mix Engineer. You are provided with:
    1. Metadata: Stem groupings (e.g., S01, S02), individual raw stems (e.g., R01, R02) and component labels (Bass,
         Melody).
    2. Audio Stems: Raw recordings with corresponding loudness in LUFS.
    3. Reference Stereo Mix: For tonal and balance reference.

  Note on Bleed: Tracks may contain signal leakage (bleed). Your spatial decisions must be based on the intended
       source of the microphone, not the leaked signals.
  Note on Room Mics: Mute tracks intended to capture room ambience (Gain = -99). Look for keywords like 'Room', '
       Ambience', or 'Main System'. Exception: Do not mute 'Overhead' (OH) tracks if they are the primary source for
       cymbals/high-frequency percussion.
  Note on Empty tracks: Listen to the track segments and identify if empty. Apply gain_db = -99 if the track segment
       is empty.

  2. The Task

  Step 1 (Overall Big Picture): Identify the genre. Use music mixing knowledge of this genre to guide mixing.
  Step 2 (Analyze & Categorize): Evaluate the role of each stem. Identify each track as: "spotlight" / "foundation" /
       "percussion" / "accompanying_parts" / "atmosphere" / "others".
  Step 3 (Three-Stage Gain Staging): Calculate gain_db = Gain_A + Offset_B + Offset_C.
    Step A (Normalization): Gain_A = (-18) - (Raw_LUFS)
    Step B (Aesthetic Balance): Apply relative boost/cut based on role.
    Step C (Compensation Gain): Final mental audit of the summed mix.
  Step 4 (Spatial Placement): Assign each stem azimuth and elevation.
  Step 5 (Source Placement Principles):
    - Center Anchor: Lead Vocals, Bass, Kick near center (0 deg azimuth).
    - Clarity through Space: Separate competing frequencies by placing instruments apart from each other.
    - Verticality: Higher-frequency elements at higher elevations.
    - Immersive Depth: Rear quadrants for atmospheric elements.
  Step 6 (Confidence Score): Include confidence score from 0 to 1.

  5. Output Format: Reasoning and JSON.
  {"confidence_score": 0.8,
   "stems": {"S01": {"instrument": "vocal",
     "raw": {"R01": {"filename": "lead_vocal.wav", "track_role": "spotlight",
       "reasoning": "...", "azimuth": 0, "elevation": 0, "gain_db": -6.5}}}}}
```

Figure 5: Full prompt used for teacher generation and student inference.

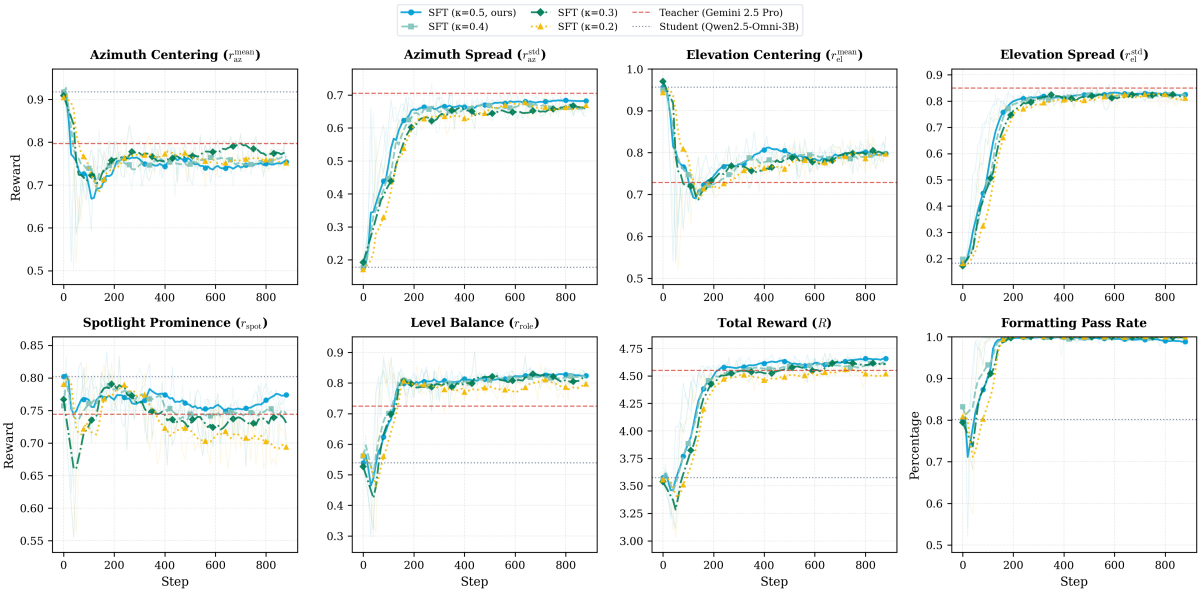

Figure 6: SFT training dynamics across filtering thresholds $\kappa \in \{0.2, 0.3, 0.4, 0.5\}$, with stricter filtering ($\kappa = 0.5$) achieving the highest total reward.