STEER UNIFIED AUDIO EDITING TOWARD RELIABLE SOURCE SELECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Unified audio editors support diverse operations, yet often struggle to modify the intended source while preserving non-target content. To strengthen this selective editing capability, we propose *Source-selection Training via Early-trajectory Emphasis and Recovery (**STEER**)*, a two-stage post-training framework, and use it to develop SteerSound, a unified audio editing model supporting six common tasks. Concretely, intermediate-state observation and bidirectional prefix interventions show that early denoising states causally influence final source selection, while incorrect early decisions often persist through subsequent denoising. Guided by these findings, STEER-Emphasis emphasizes training on task-relevant early-trajectory states to reduce source-selection errors, while STEER-Recovery learns corrective denoising from self-generated rollout states to recover from early-trajectory deviations. We further introduce the Selective Audio Editing Benchmark, enabling large-scale automatic evaluation of target manipulation and non-target preservation. Experiments on both the benchmark and more realistic multi-source audio scenes show that STEER consistently improves selective editing reliability, reducing source-selection errors while improving overall editing performance. Further evaluations across different models and other selective-control tasks also support the board generalizability of STEER and its underlying early-trajectory training rationale. SteerSound achieves advanced performance among open-source unified editors.

1 INTRODUCTION

Recent advances in diffusion and Flow Matching have enabled unified audio editors to follow diverse instructions and perform multiple editing operations within a single model Wang et al. (2023); Tao et al. (2025); Tian et al. (2026); Li et al. (2026b). However, reliable unified editing requires more than broad task coverage. Editing tasks like removal, replacement, and pitch or volume modification require the model to identify the intended source, apply the requested operation, and preserve the remaining content. We refer to these as selective editing tasks. Our experiments show that even strong existing models, like UNISON Li et al. (2026b) and SAM Audio Shi et al. (2025), can leave the intended source unchanged, modify a non-target source, or alter both, as illustrated in Figure 1 (left). These selective editing failures highlight the need to strengthen not only audio generation, but also the model's ability to select the correct editing source.

To understand these failures, we investigate how source selection develops along the generation trajectory. By analyzing the editing tendency at different denoising steps, we find that incorrect source-selection tendencies form early in the generation trajectory. This suggests that establishing the correct source selection early may be critical to the final editing outcome. Thus, we further conduct bidirectional early-trajectory interventions: replacing the early trajectories of failed edits with correct ones, and those of successful edits with deliberately incorrect ones. As shown in Figure 1 (middle), across UNISON and SAM Audio, correct early trajectories can rescue failed edits, whereas incorrect ones can induce source-selection failures, showing that early source selection strongly shapes which source is ultimately edited.

Together, these experiments reveal a common pattern across selective editing tasks: the source to be edited is progressively established during the early generation trajectory. Once an incorrect source is selected, later denoising tends to follow the resulting trajectory rather than revise the earlier de-

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.

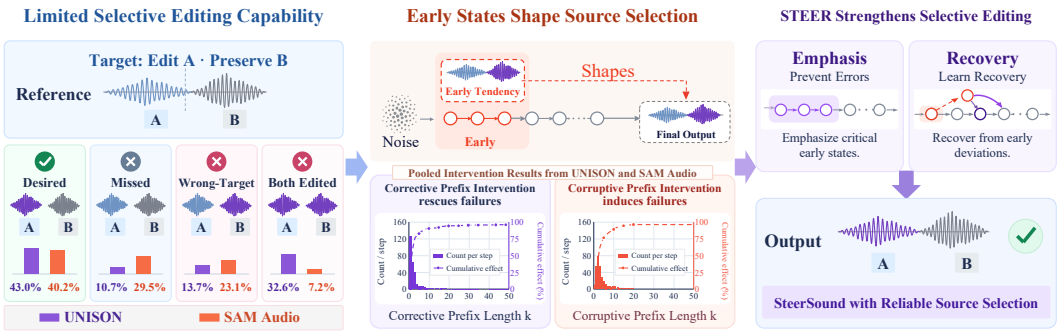

Figure 1: Strengthening selective audio editing with STEER. Left: Existing editors may leave the target unchanged or modify non-target content. Middle: Early states causally shape final source selection, and later denoising tends to preserve their editing direction through to the output rather than correct early errors. Right: STEER addresses these failures through early-trajectory emphasis (Focus) and recovery training (Recover), yielding SteerSound with more reliable selective editing.

cision. These findings motivate us to address this problem from two complementary directions: preventing source-selection errors from forming early and improving recovery when they occur. We therefore propose *Source-selection Training via Early-trajectory Emphasis and Recovery (STEER)*, a two-stage post-training framework that strengthens source selection in pretrained audio editors. As shown in Figure 1 (right), STEER-Emphasis places greater training emphasis on critical early-trajectory states to reduce the formation of incorrect source selections. Building on this, STEER-Recovery learns corrective denoising from self-generated rollout states to recover from early trajectory deviations.

To enable more comprehensive and systematic evaluation, we first introduce the Selective Audio Editing Benchmark for large-scale assessment of source-selection reliability. Moreover, existing unified audio editors remain limited in selective task coverage, making broad analysis across diverse editing operations difficult. We therefore develop SteerSound-Base, an extensively pretrained unified editor covering six generation and editing tasks, including four selective editing tasks, providing a stronger and more comprehensive basis for cross-task analysis. Applying STEER to SteerSound-Base yields SteerSound, which substantially improves source-selection reliability while achieving advanced editing performance among open-source unified audio editors. Beyond SteerSound, STEER also transfers effectively to UNISON and SAM Audio for selective editing, while early-trajectory emphasis further extends to target speaker extraction-related tasks, supporting the broader generalizability of our approach.

Our contributions are summarized as follows. **First**, through intermediate-state analysis and bidirectional prefix interventions, we show that source-selection tendencies emerge early and that early states shape final source selection in unified audio editing. **Second**, we propose STEER, a two-stage post-training framework for reliable source selection: STEER-Emphasis prioritizes critical early-trajectory states, while STEER-Recovery learns corrective denoising from self-generated rollout states. **Third**, STEER improves selective and overall editing performance on SteerSound-Base and generalizes across different model architectures, while its underlying early-trajectory training rationale also extends to related selective-control tasks with substantial improvements. **Finally**, we introduce the Selective Audio Editing Benchmark, enabling large-scale automatic evaluation of source-level editing success and systematic measurement of selective editing reliability.

2 Related Work

Unified and Instruction-Guided Audio Editing. Text-guided audio editing has been explored through training-based and training-free diffusion approaches, including AUDIT Wang et al. (2023), ZETA Manor & Michaeli (2024), PPAE Xu et al. (2024), and AudioEditor Jia et al. (2025), while recent systems such as MMEdit Tao et al. (2025), Audio-Omni Tian et al. (2026), and UNISON Li et al. (2026b) further broaden task coverage and integrate audio-language modeling with audio editing. However, source-selection reliability in unified audio editors remains underexplored. We develop SteerSound, a unified six-task audio editor, and strengthen this selective editing capability.

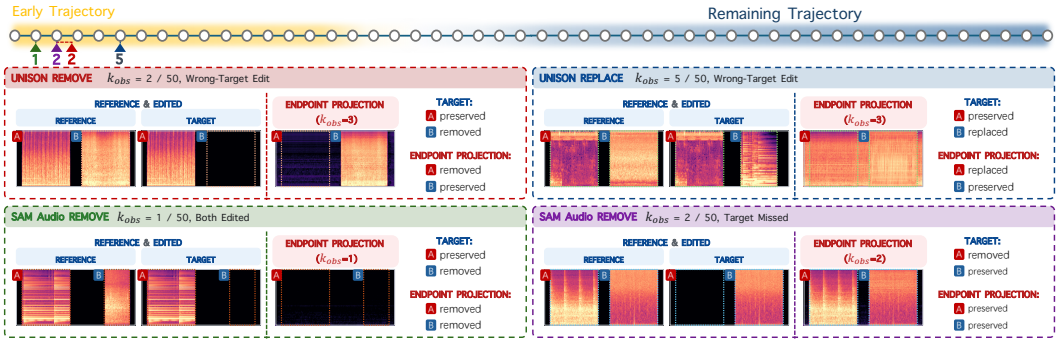

Figure 2: Early emergence of source-selection tendencies. We extrapolate intermediate denoising states to estimated zero-noise outputs and representative failed edits from different selective tasks across models are shown; Erroneous source-selection tendencies become perceptually distinguishable early in the generation trajectory, with k_{obs} marking the earliest such step.

Denoising Dynamics for Source Selection. Prior work has shown that denoising timesteps can play non-uniform roles, through stage-specialized denoisers Balaji et al. (2022), timestep reweighting or sampling Hang et al. (2023); Esser et al. (2024), and task-dependent timestep emphasis in diffusion-based visual perception Pang et al. (2025). Related studies also use model-generated states to reduce train–inference mismatch or trajectory errors, such as Self Forcing Huang et al. (2026) for autoregressive video diffusion and SOAR Qin et al. (2026) for diffusion post-training. However, these works do not study how an audio editor selects which source to modify, nor how such source-selection decisions evolve along the denoising trajectory. We show that source selection is established along a task-dependent early trajectory, motivating greater training emphasis on critical early states and recovery from early deviations.

3 Source Selection Along the Trajectory

To strengthen source selection in unified audio editing, we examine how source-selection decisions evolve during generation. We first investigate when source-selection tendencies emerge and then test the role of early-trajectory states in final source selection on UNISON Li et al. (2026b), a unified audio editor, and SAM Audio Shi et al. (2025), which we evaluate for source removal.

Flow Matching and trajectory notation. Given a clean target latent $\mathbf{z}_0$ and Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, Flow Matching Lipman et al. (2022) constructs $\mathbf{z}_\sigma = (1-\sigma)\mathbf{z}_0 + \sigma\boldsymbol{\epsilon}$ and optimizes

$$\mathcal{L}_{\text{FM}} = \|f_\theta(\mathbf{z}_\sigma, \sigma, c) - (\boldsymbol{\epsilon} - \mathbf{z}_0)\|_2^2, \tag{1}$$

where c denotes the editing condition. We denote the denoising schedule by $1 = \sigma_0 > \sigma_1 > \cdots > \sigma_K = 0$ and the state after k steps by $\mathbf{z}^{(k)}$, with $\mathbf{z}^{(k)} = \mathbf{z}^{(k-1)} + (\sigma_k - \sigma_{k-1})\mathbf{v}_{k-1}$. Unless otherwise specified, we use $K = 50$ denoising steps.

3.1 Emergence of Source-Selection Tendencies

To examine when source-selection tendencies emerge, we inspect the editing direction represented at intermediate stages of the model's denoising trajectory. Since intermediate states remain noisy, we estimate their editing direction by extrapolating each state to the zero-noise endpoint using its instantaneous velocity prediction:

$$\hat{\mathbf{z}}_0^{(k)} = \mathbf{z}^{(k)} - \sigma_k \mathbf{v}_k, \tag{2}$$

where $\mathbf{v}_k = f_\theta(\mathbf{z}^{(k)}, \sigma_k, c)$. Decoding $\hat{\mathbf{z}}_0^{(k)}$ reveals the source-selection tendency at step k, allowing us to inspect which source the model tends to edit and which it tends to preserve. This allows us to identify when an erroneous source-selection tendency begins to form during generation. We apply this probe to failed examples from selective editing tasks across the two models. As shown in Figure 2, erroneous source-selection tendencies are already formed and perceptually discernible in the early trajectory, even though the underlying states remain noisy.

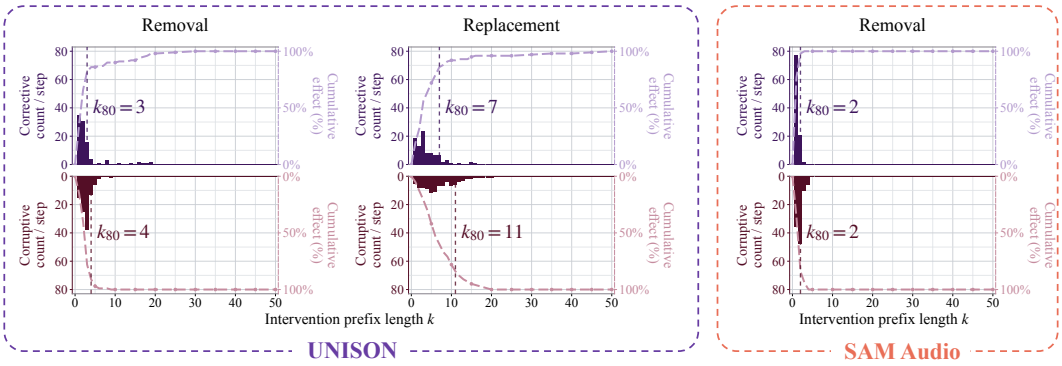

Figure 3: Bidirectional early-trajectory interventions on source selection. Corrective (top) and corruptive (bottom) prefix interventions are evaluated on 100 validation examples per task. Bars show per-step corrective or corruptive counts, while curves show the corresponding cumulative effects. For each task, k_{80} denotes the shortest intervention prefix length at which the cumulative effect reaches 80%. Removal consistently exhibits a shorter k_{80} than replacement.

3.2 Early Trajectories Shape Final Source Selection

To test whether early trajectory states causally influence final source selection, we perform corrective and corruptive prefix interventions with varying lengths k. For failed edits, we replace the first k model-predicted velocities with oracle velocities constructed from the correct editing target. For successful edits, we instead replace the first k velocities with those constructed from a deliberately corrupted target encoding incorrect source selection. In both cases, the model resumes its own denoising under the original instruction.

As shown in Figure 3, short corrective prefixes can rescue failed edits, whereas short corruptive prefixes can induce source-selection failures. Across the evaluated tasks and models, extending these prefixes into later denoising stages yields limited additional effects. These results provide evidence that early trajectory states shape final source selection, while subsequent denoising often follows the editing direction established early, even when it is incorrect. Two properties of standard training may partly explain this behavior. SD3-style logit-normal sampling Esser et al. (2024) emphasizes intermediate noise levels, leaving high-noise early states relatively underrepresented. Meanwhile, training conditions on noisy versions of correct targets, whereas inference states are self-generated, which may make recovery from early source-selection errors difficult.

Different editing tasks exhibit different sensitivities to early prefix replacement, reflecting differences among editing operations. Removal sharply contrasts target suppression with non-target preservation, making source selection apparent within only a few steps, whereas replacement requires more steps for the modification to become distinguishable, yielding broader but still early critical ranges. We therefore group operations into *generation-oriented*, *selective suppression*, and *selective transformation* tasks, depending on whether they synthesize new content, apply strongly contrasting operations to the target and preserved sources, or modify the target's content or acoustic attributes.

4 Unified Audio Editing with STEER

Building on the findings in Section 3, we develop SteerSound-Base to extend the trajectory analysis across diverse editing operations within a unified model. We then use the resulting task-dependent critical ranges to guide STEER-Emphasis, while STEER-Recovery learns to correct self-generated trajectory deviations.

4.1 SteerSound-Base

To guide source-selection training in unified audio editors, we study how early-trajectory dynamics vary across editing tasks. While existing editors typically support only a subset of the selective operations considered here, we develop SteerSound-Base to enable broader cross-task analysis within a single model. It supports six tasks: generation and addition (generation-oriented), removal (selective suppression), and replacement, pitch modification, and volume modification (selective transforma-

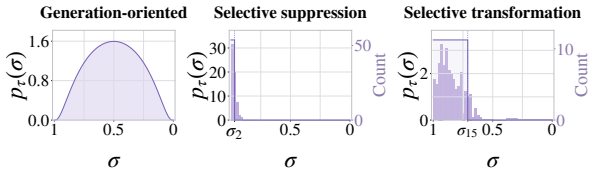

Figure 4: Noise-level sampling distributions for different task groups in STEER-Emphasis.

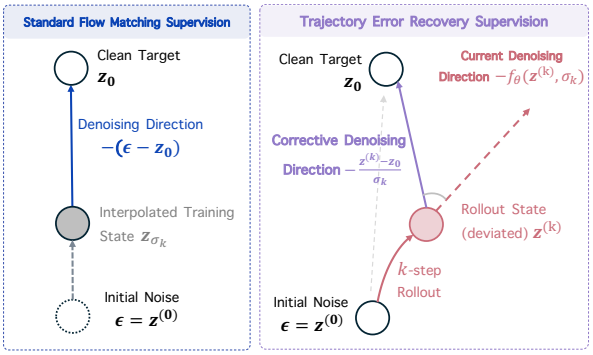

Figure 5: Standard Flow Matching versus TER supervision on self-generated rollout states.

Algorithm 1: STEER Post-Training

Input: Model θ_0; data $\mathcal{D}_{\text{tr}}, \mathcal{D}_{\text{val}}$; p_τ, K, P
Output: Post-trained model $\theta^\star$
Stage I: STEER-Emphasis
Train θ_0 with $\mathcal{L}_{\text{FM}}$ and $\sigma \sim p_\tau$
$\theta^\star \leftarrow$ validation-best checkpoint
Stage II: STEER-Recovery
$k \leftarrow 1, \quad q \leftarrow 0$
while $q < P$ **and** $k < K$ **do**
  $\theta \leftarrow \theta^\star$; reset optimizer and schedule
  for each training update **do**
    Sample $(\mathbf{z}_0, c, \tau) \sim \mathcal{D}_{\text{tr}}$
    Sample $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
    if $\tau \in \{\text{Generate}, \text{Add}\}$ **then**
      $\sigma \sim p_{\text{LN}}$; $\mathcal{L} \leftarrow \mathcal{L}_{\text{FM}}$
    else
      $\mathbf{z}^{(k)} \leftarrow \text{sg}(\text{ROLLOUT}(\theta, \boldsymbol{\epsilon}, c, k))$ ▷ no CFG
      $\mathcal{L} \leftarrow \mathcal{L}_{\text{TER}}^{(k)}$
    end if
    Update θ using $\mathcal{L}$
  end for
  $\hat{\theta}_k \leftarrow$ validation-best checkpoint at depth k
  if $S_{\text{val}}(\hat{\theta}_k) > S_{\text{val}}(\theta^\star)$ **then**
    $\theta^\star \leftarrow \hat{\theta}_k, \quad q \leftarrow 0$
  else
    $q \leftarrow q + 1$
  end if
  $k \leftarrow k + 1$
end while
return $\theta^\star$

tion). We pretrain the model on large-scale audio editing data to help reduce potential confounding from limited editing capability when estimating critical early-trajectory ranges. Architecture and pretraining details are provided in Section 5.1 and Appendix B, respectively.

4.2 Early-Trajectory Emphasis

SteerSound-Base exhibits a pattern consistent with Section 3.2: early trajectory states shape final source selection, and incorrect early selections often persist through subsequent denoising. Detailed observation and bidirectional intervention results across its four selective tasks are provided in Appendix A. To reduce the formation of these early errors, we introduce STEER-Emphasis, which concentrates post-training on states critical to source selection.

We determine task-group-specific critical ranges using the cumulative corrective effect of prefix interventions on SteerSound-Base. For a selective task group $\mathcal{G}$ and threshold ρ, we define $k_{\mathcal{G}}^{\rho} = \max_{\tau \in \mathcal{G}} \min\{k : C_\tau(k) \geq \rho\}$, where $C_\tau(k)$ denotes the cumulative corrective effect of a k-step prefix intervention on validation examples for task τ. Taking the maximum ensures that the group range covers each task's threshold-defined critical prefix. With $\rho = 0.8$, the resulting ranges span the first 2 of 50 denoising steps for selective suppression and the first 15 for selective transformation.

During post-training, we sample noise levels uniformly over the corresponding intervals for selective tasks, while retaining logit-normal sampling for generation and addition:

$$p_\tau(\sigma) = \begin{cases} p_{\text{LN}}(\sigma), & \tau \in \{\text{Generate}, \text{Add}\}, \\ \mathcal{U}(\sigma_2, 1), & \tau = \text{Remove}, \\ \mathcal{U}(\sigma_{15}, 1), & \tau \in \{\text{Replace}, \text{Pitch}, \text{Volume}\}. \end{cases} \tag{3}$$

Here p_{LN} denotes the SD3-style logit-normal distribution Esser et al. (2024), defined by $\sigma = \text{sigmoid}(u)$ with $u \sim \mathcal{N}(0, 1)$. STEER-Emphasis changes only the noise-level sampling distribution, leaving the Flow Matching objective unchanged. Figure 4 shows these distributions alongside corrective counts measured on SteerSound-Base at different prefix lengths k for the different selective task groups.

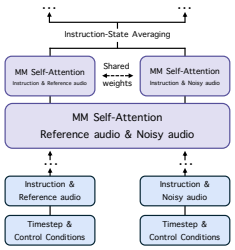

Figure 6: The architecture of SteerSound-Base.

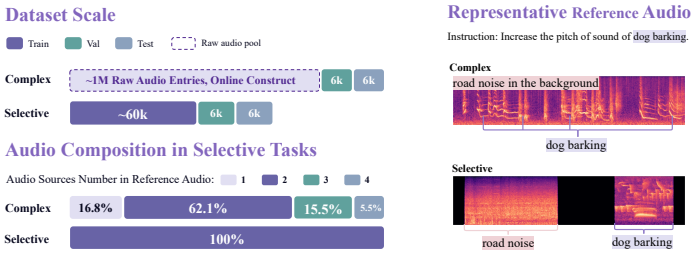

Figure 7: Statistics and representative examples of the two data settings, including dataset scale and source composition.

4.3 Early-Trajectory Recovery

STEER-Emphasis reduces early trajectory errors but cannot eliminate them entirely. We therefore introduce STEER-Recovery, which trains the model to correct early deviations and restore the intended source selection.

Recovery training starts from the validation-best checkpoint obtained after STEER-Emphasis. For each selective editing example, the current model performs a k-step denoising rollout from Gaussian noise $\boldsymbol{\epsilon}$ to obtain $\mathbf{z}^{(k)}$, without backpropagating through the rollout. We then update the model using the Trajectory Error Recovery (TER) loss:

$$\mathcal{L}_{\mathrm{TER}}^{(k)} = \left\| f_\theta(\mathbf{z}^{(k)}, \sigma_k, c) - \mathrm{sg}\left(\frac{\mathbf{z}^{(k)} - \mathbf{z}_0}{\sigma_k} \right) \right\|_2^2, \tag{4}$$

where sg denotes stop-gradient. When $\mathbf{z}^{(k)}$ lies on the target–noise interpolation path, the recovery target reduces to the standard Flow Matching target $\boldsymbol{\epsilon} - \mathbf{z}_0$. Otherwise, TER provides corrective supervision for denoising from the current state toward the desired clean edit. The model thus learns to recover from its own trajectory deviations. Figure 5 provides a schematic comparison of standard Flow Matching supervision and TER.

We progressively increase the recovery depth k from 1, starting each depth from the best checkpoint found so far. Generate and Add retain the standard Flow Matching objective throughout this stage. Training stops after P consecutive recovery depths without improvement in the best validation score, and the best checkpoint is retained. Algorithm 1 summarizes the complete STEER procedure. For our unified editor, applying both stages to SteerSound-Base produces the final SteerSound model.

5 Experiments

We first describe the experimental configuration and compare unified audio editing performance across models. We then evaluate STEER's effectiveness, conduct ablation studies on its two post-training stages, and examine its robustness to alternative definitions of the critical early-trajectory range. Next, we assess STEER's generalizability across model architectures for audio editing. Finally, we investigate whether early-trajectory emphasis also benefits related selective-control tasks.

5.1 Experimental Configuration

Base editor. As shown in Figure 6, SteerSound-Base adopts a SeedEdit-inspired dual-branch DiT Shi et al. (2024). The reference and generated audio branches exchange information through multimodal self-attention with shared attention parameters, and both receive timestep and task-specific conditioning. The model combines a 673.35M-parameter trainable backbone with a T5-Base Raffel et al. (2020) text encoder, an audio VAE Rombach et al. (2022) trained from scratch, and a HiFi-GAN Kong et al. (2020) vocoder. Further training details are provided in Appendix B.

Data, Benchmark, and Metrics. We use two complementary data settings. To better approximate real-world audio editing scenarios, we construct Complex Audio Editing Data with diverse multi-source compositions. However, source overlap in these scenes makes selective editing failures difficult to identify automatically. To support systematic analysis and large-scale experiments, we introduce the Selective Audio Editing Benchmark. Its deliberately designed source composition

Table 1: Comparison of unified audio editing performance across models on Complex Audio Editing Test Set, with task-wise metrics and parameter counts.

	Params.		Generate		Add		Replace		Remove			Pitch			Volume		
Model	Total	Train.	CLAP↑	KL↓	CLAP↑	KL↓	CLAP↑	KL↓	CLAP↑	LSD↓	KL↓	MAE↓	LSD↓	KL↓	MAE↓	LSD↓	KL↓
Audio-Omni	7.907B	3.047B	0.231	2.057	0.196	1.955	0.154	2.561	0.001	1.695	2.803	–	–	–	–	–	–
MMEdit	9.057B	0.616B	–	–	0.159	1.882	0.085	2.706	0.012	2.060	2.155	–	–	–	14.043	1.042	0.427
AUDIT	1.067B	0.860B	–	–	0.151	1.417	0.082	1.897	0.038	1.603	2.691	–	–	–	–	–	–
PPAE	1.318B	–	–	–	0.182	1.566	0.094	1.501	0.044	1.539	1.384	–	–	–	–	–	–
SDEdit	1.118B	–	–	–	0.111	2.347	0.129	2.222	0.129	1.778	1.969	0.923	0.942	1.963	5.047	0.895	1.992
ZETA	1.118B	–	–	–	0.135	2.060	0.140	1.939	0.135	1.688	1.690	0.570	0.803	1.441	4.199	0.722	1.432
SAM Audio(Small)	0.714B	0.492B	–	–	–	–	–	–	0.059	1.514	1.726	–	–	–	–	–	–
SAM Audio(Large)	3.154B	2.929B	–	–	–	–	–	–	0.056	1.465	1.629	–	–	–	–	–	–
UNISON	8.086B	0.732B	**0.247**	**1.845**	0.250	**1.218**	0.222	1.783	0.112	1.576	1.564	–	–	–	–	–	–
SteerSound-Base	1.348B	0.673B	0.223	1.915	0.249	1.286	0.230	1.032	0.137	1.346	1.140	0.055	0.493	0.281	0.702	0.433	0.200
SteerSound	1.348B	0.673B	0.220	2.133	**0.251**	1.352	**0.235**	**1.028**	**0.144**	**1.164**	**0.788**	**0.047**	**0.491**	**0.280**	**0.634**	**0.433**	**0.200**

Note: "–" in the *Train.* column denotes training-free methods built upon existing pretrained audio generation models; n/r indicates parameter counts not reported here.

allows target manipulation and non-target preservation to be evaluated separately, enabling automatic identification and measurement of selective editing failures. We also construct training and validation sets from the same distribution. Figure 7 presents statistics and representative examples.

On the complex test set, we report task-aware CLAP Score, pitch and volume MAE, LSD Wang et al. (2023), and KL-softmax Liu et al. (2023). To summarize overall unified editing performance, we introduce the Task-Balanced Gain ($\mathrm{TBG}_{\mathrm{all}}$) to measure a model's overall unified editing capability, and $\mathrm{TBG}_{\mathrm{sel}}$ to specifically quantify its selective editing capability. On the selective audio editing benchmark, we develop an evaluation protocol that measures overall editing success while identifying and categorizing different types of source-selection errors. Further details on the two datasets and their corresponding metrics are provided in Appendices C and D respectively.

Implementation Details. We use the critical early-trajectory ranges defined in Section 4.2 for STEER-Emphasis and set the patience hyperparameter to $P = 2$ for STEER-Recovery. Unless otherwise specified, all SteerSound variants use classifier-free guidance (CFG) at inference, with guidance scales of 7.5 for source addition and 3.5 for the other tasks, while other models retain their official default configurations. Experiment-specific hyperparameters, training procedures, and learning dynamics are provided in the corresponding appendices.

5.2 Unified Audio Editing Performance

We compare SteerSound with Audio-Omni Tian et al. (2026), UNISON Li et al. (2026b), SAM Audio Shi et al. (2025), MMEdit Tao et al. (2025), PPAE Xu et al. (2024), ZETA Manor & Michaeli (2024), AUDiT Wang et al. (2023), SDEdit Meng et al. (2021) on the Complex Audio Editing Test Set, as summarized in Table 1. SteerSound demonstrates superior performance on replacement, removal, pitch modification, and volume modification, and achieves the best overall performance among the compared models.

5.3 Effectiveness and Robustness of STEER

Effectiveness of STEER.

We separately post-train SteerSound-Base with STEER on the Complex and Selective Audio Editing training sets and evaluate the resulting models on their respective test sets. On the Complex test set, full STEER substantially improves overall editing performance, as shown in Table 2 A1. To further assess selective editing reliability in complex scenes, we manually annotate 100 randomly sampled test examples and categorize their source-selection errors. The human evaluation shows that STEER increases the overall success rate and markedly reduces selective editing failures (Figure 8 A1).

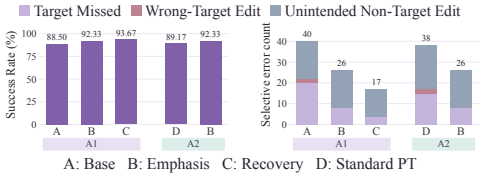

Figure 8: Human evaluation on 100 randomly sampled test examples per task corresponding to A1 and A2 in Table 2.

On the Selective Audio Editing Benchmark, STEER-Emphasis increases the overall success rate and reduces selective editing failures, while full STEER further improves both measures (Figure 9 A1).

Table 2: Ablation results of STEER within the SteerSound family on the Complex Audio Editing Test Set under different training strategies.

Exp.	Setting	Generate		Add		Replace		Remove			Pitch			Volume			Overall	Selective
		CLAP↑	KL↓	CLAP↑	KL↓	CLAP↑	KL↓	CLAP↑	LSD↓	KL↓	MAE↓	LSD↓	KL↓	MAE↓	LSD↓	KL↓	TBG_{all} ↑	TBG_{sel} ↑
A1	Base	**0.223**	**1.915**	0.249	**1.286**	0.230	1.032	0.137	1.346	1.140	0.055	0.493	0.281	0.702	**0.433**	**0.200**	0.000	0.000
	Emphasis	0.217	2.121	**0.252**	1.325	0.234	1.049	0.139	1.217	0.898	0.050	**0.491**	**0.279**	0.708	**0.433**	0.203	1.672	2.969
	Recovery	0.220	2.133	0.251	1.352	**0.235**	**1.028**	**0.144**	**1.164**	**0.788**	**0.047**	**0.491**	0.280	**0.634**	**0.433**	**0.200**	**5.358**	**8.249**
A2	Standard PT	**0.225**	**1.991**	**0.254**	**1.311**	**0.235**	1.078	**0.140**	1.325	1.057	0.054	0.492	**0.273**	0.726	**0.433**	**0.202**	0.924	0.656
	Emphasis	0.217	2.121	0.252	1.325	0.234	**1.049**	0.139	**1.217**	**0.898**	**0.050**	**0.491**	0.279	**0.708**	**0.433**	0.203	**1.672**	**2.969**
A3	$Emphasis_{15}$	0.216	2.143	0.249	1.347	**0.234**	1.057	**0.140**	1.246	0.952	**0.048**	**0.491**	**0.275**	0.742	**0.433**	**0.200**	1.363	2.816
	Emphasis	**0.217**	**2.121**	**0.252**	**1.325**	**0.234**	**1.049**	0.139	**1.217**	**0.898**	0.050	**0.491**	0.279	**0.708**	**0.433**	0.203	**1.672**	**2.969**
A4	$Emphasis_{cont}$	0.219	**2.076**	**0.253**	**1.318**	**0.235**	1.037	0.136	1.250	0.933	**0.047**	**0.491**	**0.277**	0.681	**0.433**	**0.200**	3.607	5.499
	Recovery	**0.220**	2.133	0.251	1.352	**0.235**	**1.028**	**0.144**	**1.164**	**0.788**	**0.047**	**0.491**	0.280	**0.634**	**0.433**	**0.200**	**5.358**	**8.249**

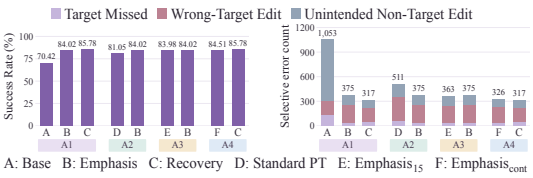

Figure 9: Success rates and selective error distributions on the Selective Audio Editing Benchmark under different training strategies.



Figure 10: Overall success rates of UNISON and SAM Audio on the Selective Audio Editing Benchmark under different training strategies.

Together, these results demonstrate that STEER improves both overall editing performance and source-selection reliability. More details are provided in Appendix E.1.

STEER-Emphasis Ablation with Standard PT. To isolate the benefit of early-trajectory emphasis from that of additional training, we compare STEER-Emphasis with Standard PT, which follows the standard Flow Matching post-training protocol Esser et al. (2024). Both use the same training settings and number of epochs, differing only in the noise-level sampling distribution. We report automatic evaluation on the complex test set in Table 2 A2, the corresponding human evaluation in Figure 8 A2, and results on the Selective Audio Editing Benchmark in Figure 9 A2. Across both data settings, STEER-Emphasis achieves better overall editing performance and higher success rates than Standard PT, while reducing source-selection errors. These results demonstrate the benefit of emphasizing critical early-trajectory states beyond standard post-training. More details are provided in Appendix E.2.

Range Robustness of STEER-Emphasis.

To test whether STEER-Emphasis depends on the specific critical early-trajectory range definition, we replace the original 80% cumulative-corrective-effect criterion with an elbow-based rule. At each intervention depth k, we average the corrective count and cumulative corrective effect across the four selective tasks, as shown in Figure 11. We select the elbow of the averaged corrective-count curve Satopaa et al. (2011) as a unified boundary, yielding $k = 15$.

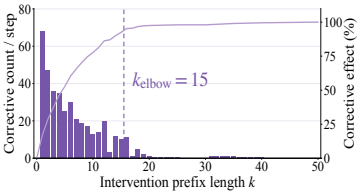

Figure 11: Average corrective effect and count, with the elbow at $k = 15$.

The resulting variant, $STEER\text{-}Emphasis_{15}$, samples training noise levels uniformly from $[\sigma_{15}, 1]$ for all selective tasks. We compare it with the original task-group-specific STEER-Emphasis, keeping all other training settings and hyperparameters unchanged. Results on the complex test set and the Selective Audio Editing Benchmark are reported in Table 2 A3 and Figure 9 A3, respectively. The two variants achieve comparable overall performance on both test sets, demonstrating that the effectiveness of STEER-Emphasis is not tied to the original threshold-based range definition. More details are provided in Appendix E.3.

STEER-Recovery Ablation with Matched Updates. To isolate the benefit of recovery training from that of additional optimization, we continue STEER-Emphasis training with the same number of parameter updates as the selected STEER-Recovery checkpoint, yielding a continuation baseline

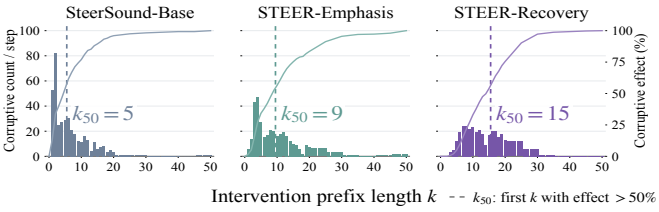

Figure 12: Recovery from corrupted early trajectories across SteerSound variants on Selective Audio Editing Benchmark.

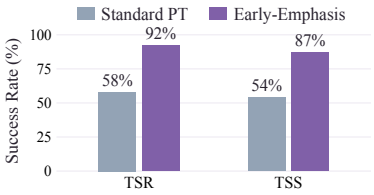

Figure 13: Human evaluation on TSE-related tasks under different training strategies.

denoted as STEER-Emphasis$_{\text{cont}}$. Both runs start from the same checkpoint, and we match their training hyperparameters as closely as possible. Results on the complex test set and the Selective Audio Editing Benchmark are reported in Table 2 A4 and Figure 9 A4, respectively. Across both data settings, STEER-Recovery achieves better overall editing performance and more reliable source selection than this continuation baseline. These results support the benefit of explicitly learning recovery from self-generated trajectory states beyond simply extending STEER-Emphasis training. More details are provided in Appendix E.4.

Recovery after Early Corruption. We further evaluate whether STEER-Recovery improves the model's ability to recover from early-trajectory errors. We apply the same corruptive early-trajectory intervention protocol described in Section 3.2 to the Selective Audio Editing Benchmark. As shown in Figure 12, the 50% corruptive-effect point shifts from $k_{50} = 5$ for SteerSound-Base to $k_{50} = 9$ after STEER-Emphasis and $k_{50} = 15$ after full STEER. This shift indicates that longer corrupted prefixes are required to reach the same corruptive-effect threshold, supporting the effectiveness of STEER-Recovery in improving robustness to early-trajectory errors and recovery during subsequent denoising. More details are provided in Appendix E.5.

5.4 Generalization of STEER

We apply STEER to UNISON and SAM Audio Shi et al. (2025) for unified audio editing and source removal, respectively. As shown in Figure 10, STEER-Emphasis achieves higher success rates than Standard PT on both models, while STEER-Recovery yields further improvements over STEER-Emphasis$_{\text{cont}}$ with the same number of parameter updates. Additional results in Appendix F provide further evidence of improved selective editing capability. Together, these findings demonstrate that STEER transfers effectively across model architectures for audio editing.

5.5 Transferability to Related Selective-Control Tasks

We further examine whether early-trajectory emphasis extends beyond category-level audio editing to related selective-control tasks. To this end, we study transcript-conditioned target speaker extraction (TSE)-related tasks, adapting SteerSound-Base for speaker removal and SAM Audio for separation. These tasks require distinguishing competing speakers by speech content rather than sound category. We compare standard Flow Matching training with early-trajectory emphasis from the start of task adaptation. The quantitative results in Table 3 and human evaluation on 100 randomly sampled test examples per task in Figure 13 show substantial gains on both tasks, with early-trajectory emphasis enabling effective target selection where standard training struggles. These findings support the transferability of the early-trajectory training principle to learning fine-grained source-selection capabilities. Further details are provided in Appendix G.

Table 3: Objective evaluation on TSE-related tasks under different strategies.

Method	SteerSound CLAP-AA$_{\text{TSR}}$ ↑	SAM Audio CLAP-AA$_{\text{TSS}}$ ↑
Standard PT	0.7441	-0.0033
STEER-Emphasis	**0.8564**	**0.2123**

6 Conclusion

We identify early trajectories as critical to source selection and introduce STEER, combining early-trajectory emphasis with recovery training. Its effectiveness across architectures highlights a transferable approach to improving both editing performance and source-selection reliability.

7 AI USE STATEMENT

Generative AI tools were used only for manuscript writing and language polishing. They were not used for research design, methodology development, data analysis, or result interpretation. All AI-assisted text was reviewed and revised by the authors, who take full responsibility for the final content.

REFERENCES

Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Qinsheng Zhang, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, et al. ediff-i: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022.

Dong Chen, Fangyun Wei, Ziyu Wan, Dongdong Chen, Jiawei Zhang, Jinjing Zhao, Sirui Zhang, Yang Yue, Zhiyang Liang, Baining Guo, et al. Lens: Rethinking training efficiency for foundational text-to-image models. *arXiv preprint arXiv:2605.21573*, 2026.

Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. Vggsound: A large-scale audio-visual dataset. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 721–725. IEEE, 2020.

Konstantinos Drossos, Samuel Lipping, and Tuomas Virtanen. Clotho: An audio captioning dataset. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 736–740. IEEE, 2020.

Benjamin Elizalde, Soham Deshmukh, Mahmoud Al Ismail, and Huaming Wang. Clap learning audio concepts from natural language supervision. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Eduardo Fonseca, Xavier Favory, Jordi Pons, Frederic Font, and Xavier Serra. Fsd50k: an open dataset of human-labeled sound events. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:829–852, 2021.

Tiankai Hang, Shuyang Gu, Chen Li, Jianmin Bao, Dong Chen, Han Hu, Xin Geng, and Baining Guo. Efficient diffusion training via min-snr weighting strategy. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7407–7417. IEEE, 2023.

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

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *Advances in Neural Information Processing Systems*, 38:167283–167308, 2026.

Yuhang Jia, Yang Chen, Jinghua Zhao, Shiwan Zhao, Wenjia Zeng, Yong Chen, and Yong Qin. Audioeditor: A training-free diffusion-based audio editing framework. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025.

Chris Dongjoo Kim, Byeongchang Kim, Hyunmin Lee, and Gunhee Kim. Audiocaps: Generating captions for audios in the wild. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 119–132, 2019.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *Advances in neural information processing systems*, 33:17022–17033, 2020.

Khaled Koutini, Jan Schlüter, Hamid Eghbal-Zadeh, and Gerhard Widmer. Efficient training of audio transformers with patchout. *arXiv preprint arXiv:2110.05069*, 2021.

Kai Li, Jintao Cheng, Chang Zeng, Zijun Yan, Helin Wang, Zixiong Su, Bo Zheng, and Xiaolin Hu. A semantically consistent dataset for data-efficient query-based universal sound separation. *arXiv preprint arXiv:2601.22599*, 2026a.

Zhaoqing Li, Haoning Xu, Jingran Su, Yaofang Liu, Zhefan Rao, Huimeng Wang, Jiajun Deng, Tianzi Wang, Zengrui Jin, Rui Liu, et al. Unison: A unified sound generation and editing framework via deep llm fusion. *arXiv preprint arXiv:2605.31530*, 2026b.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark D Plumbley. Audioldm: Text-to-audio generation with latent diffusion models. *arXiv preprint arXiv:2301.12503*, 2023.

Hila Manor and Tomer Michaeli. Zero-shot unsupervised and text-based audio editing using ddpm inversion. *arXiv preprint arXiv:2402.10009*, 2024.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021.

Irene Martin Morato and Annamaria Mesaros. Macs-multi-annotator captioned soundscapes. *MACS-Multi-Annotator Captioned Soundscapes*, 2021.

Xichen Pan, Satya Narayan Shukla, Aashu Singh, Zhuokai Zhao, Shlok Kumar Mishra, Jialiang Wang, Zhiyang Xu, Jiuhai Chen, Kunpeng Li, Felix Juefei-Xu, et al. Transfer between modalities with metaqueries. *arXiv preprint arXiv:2504.06256*, 2025.

Ziqi Pang, Xin Xu, and Yu-Xiong Wang. Aligning generative denoising with discriminative objectives unleashes diffusion for visual perception. In *International Conference on Learning Representations*, volume 2025, pp. 101991–102013, 2025.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, 2023.

You Qin, Linqing Wang, Hao Fei, Roger Zimmermann, Liefeng Bo, Qinglin Lu, and Chunyu Wang. Soar: Self-correction for optimal alignment and refinement in diffusion models. *arXiv preprint arXiv:2604.12617*, 2026.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10674–10685. ieee, 2022.

Ville Satopaa, Jeannie Albrecht, David Irwin, and Barath Raghavan. Finding a" kneedle" in a haystack: Detecting knee points in system behavior. In *2011 31st international conference on distributed computing systems workshops*, pp. 166–171. IEEE, 2011.

Kentaro Seki, Nobutaka Ito, Kazuki Yamauchi, Yuki Okamoto, Kouei Yamaoka, Yuki Saito, Shinnosuke Takamichi, and Hiroshi Saruwatari. Language-queried target speech extraction using paralinguistic and non-linguistic prompts. *Acoustical Science and Technology*, pp. e25–27, 2025.

Bowen Shi, Andros Tjandra, John Hoffman, Helin Wang, Yi-Chiao Wu, Luya Gao, Julius Richter, Matt Le, Apoorv Vyas, Sanyuan Chen, et al. Sam audio: Segment anything in audio. *arXiv preprint arXiv:2512.18099*, 2025.

Yichun Shi, Peng Wang, and Weilin Huang. Seededit: Align image re-generation to image editing. *arXiv preprint arXiv:2411.06686*, 2024.

Ye Tao, Wen Wu, Chao Zhang, Mengyue Wu, Shuai Wang, and Xuenan Xu. Mmedit: A unified framework for multi-type audio editing via audio language model. *arXiv preprint arXiv:2512.20339*, 2025.

Zeyue Tian, Binxin Yang, Zhaoyang Liu, Jiexuan Zhang, Ruibin Yuan, Hubery Yin, Qifeng Chen, Chen Li, Jing Lyu, Wei Xue, et al. Audio-omni: Extending multi-modal understanding to versatile audio generation and editing. *arXiv preprint arXiv:2604.10708*, 2026.

Yuancheng Wang, Zeqian Ju, Xu Tan, Lei He, Zhizheng Wu, Jiang Bian, et al. Audit: Audio editing by following instructions with latent diffusion models. *Advances in Neural Information Processing Systems*, 36:71340–71357, 2023.

Manjie Xu, Chenxing Li, Dan Su, Wei Liang, Dong Yu, et al. Prompt-guided precise audio editing with diffusion models. *arXiv preprint arXiv:2406.04350*, 2024.

Chen Yang, Chufan Yu, Hanfu Chen, Jie Zhu, Jingqi Chen, Ke Chen, Wenxuan Wang, Yang Wang, Yaozhou Jiang, Yi Jiang, et al. Moss-audio technical report. *arXiv preprint arXiv:2606.01802*, 2026.

APPENDIX CONTENTS

A Early-Trajectory Experiments on SteerSound-Base **14**

A.1 Early-Trajectory Observation 14

A.2 Bidirectional Early-Trajectory Intervention 14

B SteerSound-Base: Architecture and Training Details **14**

B.1 Task and Timestep Conditioning 15

B.2 Training Configuration 15

B.3 Task-Dependent Inference Prompts 15

C Complex Audio Editing Data: Construction and Evaluation **16**

C.1 Audio Data Sources 16

C.2 Online Data Constructor 16

C.3 Evaluation Metrics 17

D Selective Audio Editing Benchmark: Construction and Evaluation **19**

D.1 Benchmark Construction 19

D.2 Automatic Evaluation Protocol 19

E Details on the Effectiveness and Robustness of STEER **22**

E.1 Details on Effectiveness of STEER 22

E.2 Details on STEER-Emphasis Ablation with Standard PT 24

E.3 Details on Range Robustness of STEER-Emphasis 24

E.4 Details on STEER-Recovery Ablation with Matched Updates 25

E.5 Details on Recovery after Early Corruption 26

F Details on Generalization of Steer **26**

F.1 Details on the STEER Generalization to UNISON 26

F.2 Details on the STEER Generalization to SAM Audio 28

G Details on Transferability to Related Selective-Control Tasks **31**

G.1 Details on Target Speaker Removal with SteerSound-Base 31

G.2 Details on Target Speaker Separation with SAM Audio 33

H Human Evaluation and Website Design **36**

H.1 Human Annotation Procedure and Criteria 36

H.2 Website Design 36

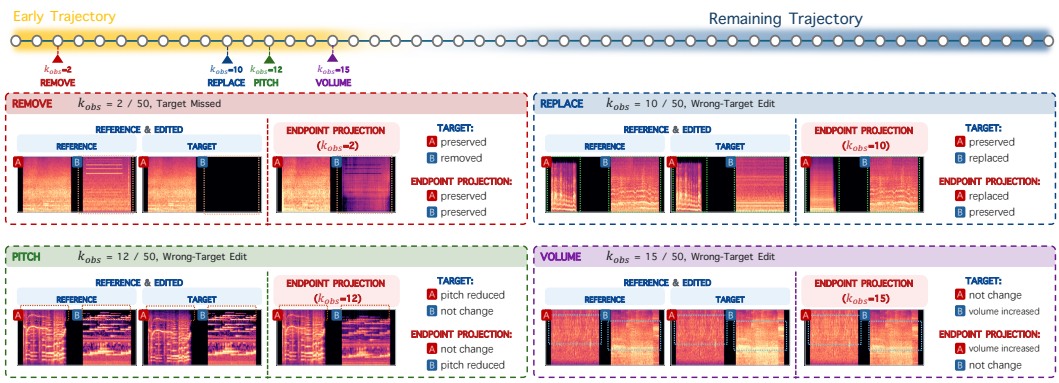

Figure 14: Early emergence of source-selection tendencies on SteerSound-Base. Intermediate denoising states are extrapolated to estimated zero-noise outputs. Representative failed edits from the four selective editing tasks are shown, with k_{obs} denoting the earliest step at which the erroneous source-selection tendency becomes perceptually distinguishable.

A Early-Trajectory Experiments on SteerSound-Base

This appendix presents the early-trajectory analyses on SteerSound-Base. While Section 3 establishes the early formation and causal influence of source selection on UNISON and SAM Audio, here we extend the same analyses to the four selective editing tasks supported by SteerSound-Base: removal, replacement, pitch modification, and volume modification. We first examine when erroneous source-selection tendencies become observable, and then use bidirectional prefix interventions to identify the corresponding source-selection-critical early ranges.

A.1 Early-Trajectory Observation

Figure 14 shows representative cases. Across all four tasks, erroneous source-selection tendencies become perceptually distinguishable during the early trajectory. Removal reveals the selected source within only a few steps, whereas replacement and source-attribute modifications require a broader, but still early, portion of the trajectory before their editing tendencies become clearly distinguishable. These observations are consistent with the task-dependent behavior discussed in Section 3.1.

A.2 Bidirectional Early-Trajectory Intervention

Following the protocol in Section 3.2, we perform the same bidirectional prefix interventions on SteerSound-Base using 100 validation examples per selective editing task. As shown in Figure 15, the intervention effects are likewise concentrated in the early trajectory, with removal exhibiting a substantially shorter sensitive range than replacement, pitch modification, and volume modification. Applying the cumulative-corrective-effect criterion defined in Section 4.2 with $\rho = 0.8$ yields $k^* = 2$ of 50 steps for selective suppression and $k^* = 15$ for selective transformation. These ranges are used by STEER-Emphasis.

B SteerSound-Base: Architecture and Training Details

This appendix complements Section 5.1 with additional implementation details of SteerSound-Base. As introduced in the main paper, SteerSound-Base serves as the common backbone for our trajectory analysis and subsequent STEER post-training. Table 4 summarizes the parameter composition of the complete system and its trainable editing backbone.

Here, we focus on the implementation details omitted from the main text, including the conditioning mechanism, base-model optimization, and task-dependent inference configuration. We

Table 4: Parameter counts by module for SteerSound-Base.

Module	**Params. (M)**
DiT backbone	673.35
T5-Base	109.63
Audio VAE	509.34
HiFi-GAN	55.26
Total	1,347.58
Trainable	673.35

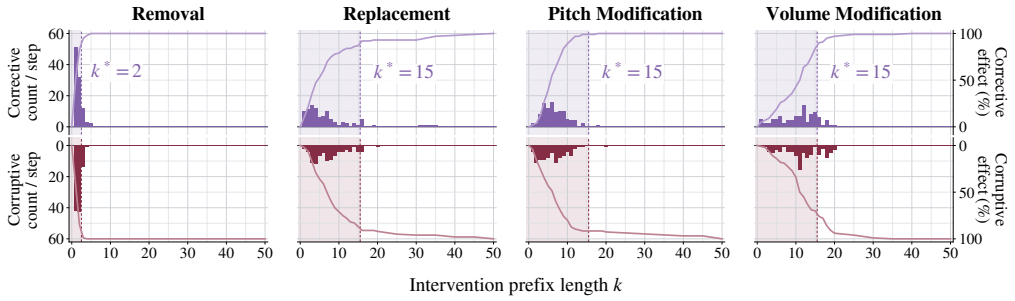

Figure 15: Bidirectional early-trajectory interventions on SteerSound-Base. Corrective (top) and corruptive (bottom) interventions are evaluated on 100 validation examples per selective editing task. Bars denote per-step intervention counts and curves denote cumulative effects. Dashed lines indicate the resulting critical ranges used by STEER-Emphasis: $k^* = 2$ for selective suppression and $k^* = 15$ for selective transformation.

first describe how task-specific scalar controls are incorporated into the denoising backbone. Then we provide the optimization and learning-rate configuration used to obtain SteerSound-Base, followed by the construction of task-dependent CFG reference prompts at inference time.

B.1 Task and Timestep Conditioning

In addition to the text instruction, the backbone receives four scalar conditions: the noise level σ, task ID τ, volume change Δ_v, and pitch change Δ_p. To combine denoising and editing information, each input is normalized to $[0, 1]$ and passed through sinusoidal encoding followed by an independent MLP. The four equal-dimensional embeddings are then summed:

$$\mathbf{c}_{\text{global}} = g_\sigma(\phi(\bar{\sigma})) + g_\tau(\phi(\bar{\tau})) + g_v\big(\phi(\overline{\Delta_v})\big) + g_p\big(\phi(\overline{\Delta_p})\big)\,, \tag{5}$$

where bars denote normalization, ϕ is sinusoidal encoding, and each g is an independent MLP. The resulting vector modulates intermediate features in both audio branches, following DiT and SD3 (Peebles & Xie, 2023; Esser et al., 2024), while T5 provides text-token conditioning separately. Notably, in our model, we set the pitch-change range Δ_p to $[-4, +4]$ semitones with a step size of 0.5 semitones, and the volume-change range Δ_v to $[-5, +5]$ dB. The signs specify the direction of the requested change.

B.2 Training Configuration

To train this conditioned backbone, we use the Flow Matching objective in Equation 1, starting from random initialization. During pre-training, we adopt the standard logit-normal noise-level sampling strategy following SD3 Esser et al. (2024). We use AdamW with $(\beta_1, \beta_2) = (0.9, 0.999)$, $\epsilon = 10^{-8}$, and zero weight decay. The learning rate reaches 10^{-4} after 1,000 warmup steps and subsequently follows a cosine decay schedule. Training uses BF16 precision and DeepSpeed ZeRO-2 for 15 epochs on our Complex Audio Editing Training Set and takes approximately 16 hours on 16 Ascend 910B NPUs. Figure 16 reports the training loss and learning-rate schedule.

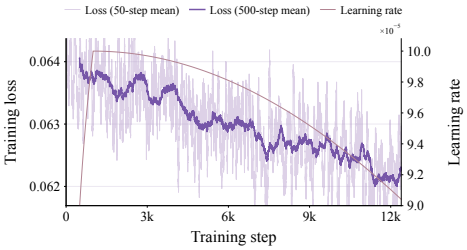

Figure 16: Training loss and learning rate.

B.3 Task-Dependent Inference Prompts

For inference, we use the CFG configuration Ho & Salimans (2022) with task-dependent reference prompts (Table 5). Generate, Add, Remove, and Replace use an empty string. In contrast, Volume and Pitch retain the direction and magnitude but omit the source description. Attribute magnitudes are absolute values with one decimal place, and Increase or Reduce specifies the direction. Units are implicit: dB for Volume and semitones for Pitch.

Table 5: Example conditional and CFG reference prompts.

Task	Conditional prompt	CFG reference prompt
Generate	Generate the sound of rain.	*Empty string*
Add	Add the sound of rain.	*Empty string*
Remove	Remove the sound of rain.	*Empty string*
Replace	Replace the sound of rain by the sound of wind.	*Empty string*
Volume ↑	Increase the volume of the sound of rain by 3.0.	Increase the volume by 3.0.
Volume ↓	Reduce the volume of the sound of rain by 3.0.	Reduce the volume by 3.0.
Pitch ↑	Increase the pitch of the sound of rain by 2.0.	Increase the pitch by 2.0.
Pitch ↓	Reduce the pitch of the sound of rain by 2.0.	Reduce the pitch by 2.0.

C Complex Audio Editing Data: Construction and Evaluation

C.1 Audio Data Sources

Task-Adaptive Caption Granularity. As described in Section 4.1, we consider six editing operations: generation, addition, removal, replacement, pitch modification, and volume modification. Through extensive experiments, we observe that different editing operations benefit from different granularities of textual supervision. Specifically, audio generation and source addition perform better with fine-grained audio descriptions, whereas source removal, pitch modification, and volume modification benefit more from class-level descriptions. This difference can be understood from their training and inference settings: fine-grained descriptions provide richer semantic constraints for generation-oriented operations, while concise class labels establish a clearer correspondence between semantic categories and audio sources, which better matches selective editing scenarios where the target source is typically specified by a category term. Similar benefits of information-rich captions have also been observed in image generation Chen et al. (2026).

Based on this observation, we use two separate source pools constructed from different types of annotated audio data: the Fine-Grained Description Pool $\mathcal{P}$ and the Class-Label Pool $\mathcal{L}$. The former consists of audio paired with fine-grained natural-language descriptions, while the latter consists of single-source audio paired with class-level labels, where each clip contains only the labeled sound source. Samples from $\mathcal{P}$ are used for audio generation, source addition, and the generated content in source replacement, whereas samples from $\mathcal{L}$ are used for source removal, pitch modification, volume modification, and the source to be replaced in source replacement.

The Fine-Grained Description Pool $\mathcal{P}$ is collected from VGGSound Chen et al. (2020), AudioCaps Kim et al. (2019), FSD50K Fonseca et al. (2021), Clotho V2 Drossos et al. (2020), and MACS Morato & Mesaros (2021). It contains 380,165 audio chunks with a total duration of 953.51 hours, with each sample paired with a fine-grained description of its acoustic content. The data composition across source datasets is summarized in Table 6. The Class-Label Pool $\mathcal{L}$ is constructed from HIVE Li et al. (2026a) by retaining audio-label pairs with CLAP similarity above 0.1. It contains 635,019 audio chunks with a total duration of 1,688.44 hours and covers 279 sound categories.

Table 6: Data sources for $\mathcal{P}$.

Dataset	# Chunks
VGGSound	192,794
AudioCaps	101,369
FSD50K	51,182
Clotho V2	17,545
MACS	17,275
Total	**380,165**

C.2 Online Data Constructor

We develop the online Complex Audio Editing Data constructor to generate training data for the overall model. For each editing task, audio sources are sampled from different source pools following the principles described in Appendix C.1, and are then composed into the reference audio, target audio, and editing instruction. The detailed training data construction process is illustrated in Fig. 17.

Since training samples are constructed online, the training set is not restricted to a fixed collection of pre-generated editing pairs. We set its nominal length to approximately 600,000 samples. We

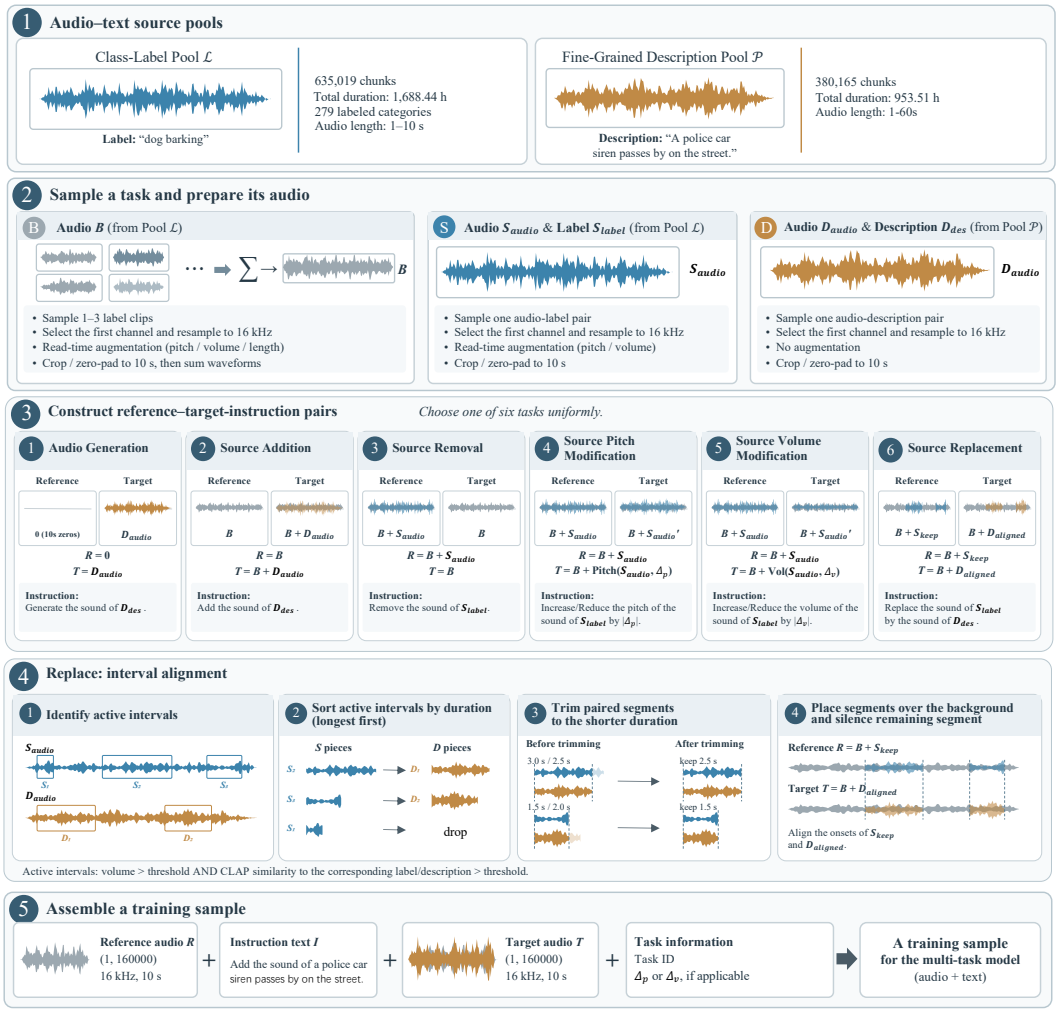

Figure 17: Overview of the online Complex Audio Editing Data constructor. Audio sources from the Class-Label Pool $\mathcal{L}$ and Fine-Grained Description Pool $\mathcal{P}$ are dynamically composed into reference–target–instruction pairs for six audio editing operations. $|\Delta_p|$ is capped at 4 semitones, and $|\Delta_v|$ is capped at 5 dB in the process.

additionally construct fixed validation and test sets, each containing 1,000 samples per operation and 6,000 samples in total. All audio sources appearing in the validation and test sets are excluded from training. Notably, during both training and evaluation, all audio samples are either cropped or padded to a fixed duration of 10 seconds.

C.3 Evaluation Metrics

We evaluate performance on Complex Audio Editing Data using task-aware semantic and attribute metrics, together with complementary audio-similarity metrics. For audio generation, source addition, source removal, and source replacement, we use a task-aware CLAP score as the primary metric. For source pitch and volume modification, the primary metrics are Pitch MAE and Volume MAE, respectively. We additionally report Log-Spectral Distance (LSD) and KL-softmax as complementary metrics.

Task-Aware CLAP. Task-aware CLAP is used to evaluate audio generation, source addition, source removal, and source replacement. These operations can naturally be characterized from three

semantic aspects: whether the sources that should be preserved remain present, whether the sources that should be added are introduced, and whether the sources that should be removed are suppressed. We therefore design the task-aware CLAP score around these three aspects. Specifically, for each sample i, let c_i^P, c_i^A, and c_i^R denote the average CLAP similarities between the generated audio and the labels corresponding to sources that should be preserved, added, and removed, respectively. The task-aware CLAP score is defined as

$$s_i^{\mathrm{CLAP}} = \begin{cases} c_i^A, & \text{Generate}, \\ c_i^P + c_i^A, & \text{Add}, \\ c_i^P - c_i^R, & \text{Remove}, \\ c_i^P + c_i^A - c_i^R, & \text{Replace}. \end{cases} \tag{6}$$

This formulation combines the semantic aspects involved in each operation to measure its overall task fulfillment. Higher values indicate better fulfillment of the task-specific semantic requirements, and the average score over all valid samples reflects the model's overall performance on the corresponding task.

Pitch and Volume MAE. For source pitch and volume modification, directly measuring the edited source would require first separating it from the generated mixture, which can introduce substantial measurement errors. We therefore evaluate these two operations indirectly by measuring the residual pitch and volume differences between the generated audio and the corresponding target audio without source separation. For each sample i, let $\widehat{\Delta}_i^{\mathrm{pitch}}$ and $\widehat{\Delta}_i^{\mathrm{volume}}$ denote the residual pitch and volume offsets, respectively. The pitch offset is estimated using bidirectional multi-resolution CQT alignment, while the volume offset is computed from the RMS-level difference between the generated and target audio. We define

$$\mathrm{MAE}_q = \frac{1}{|\mathcal{V}_q|} \sum_{i \in \mathcal{V}_q} \left| \widehat{\Delta}_i^q \right|, \qquad q \in \{\mathrm{pitch}, \mathrm{volume}\}, \tag{7}$$

where $\mathcal{V}_q$ denotes the set of valid samples for metric q. Pitch MAE is measured in semitones and Volume MAE in dB, with lower values indicating better editing performance. The task-level result is obtained by averaging over all valid samples. Unlike volume, pitch estimation depends on the measurement method, and the pitch of some audio samples cannot be measured. We therefore select the pitch estimator that provides the most accurate measurements through extensive validation of multiple estimation methods, and exclude pitch-unmeasurable samples from $\mathcal{V}_{\mathrm{pitch}}$. This measurability issue applies only to Pitch MAE.

LSD and KL-softmax. In addition to the task-related metrics above, we further evaluate the similarity between the generated and target audio from spectral and semantic perspectives. For source removal, pitch modification, and volume modification, the desired output is defined by a specific transformation of the input audio and is therefore expected to closely match the constructed target at the signal level. We thus report Log-Spectral Distance (LSD) for these three operations to measure their log-magnitude spectral discrepancy. For all six operations, we additionally report KL-softmax Liu et al. (2023), which measures the KL divergence between the PaSST Koutini et al. (2021) audio-event distributions of the target and generated audio and reflects their similarity in semantic audio-event content. Lower values indicate greater similarity for both metrics.

Overall Task-Balanced Gain. Since all six editing tasks share a single model, improvements on one task may be accompanied by degradation on others. Moreover, their evaluation metrics differ in scale and optimization direction, making direct averaging inappropriate. We therefore introduce the overall Task-Balanced Gain, denoted as $\mathrm{TBG}_{\mathrm{all}}$. Given a fixed baseline model and one primary metric for each task, let M_t and M_t^{base} denote the corresponding metrics of the evaluated and baseline models, respectively. We define

$$r_t = \begin{cases} \dfrac{M_t}{M_t^{\mathrm{base}}}, & t \in \{\mathrm{Generate}, \mathrm{Add}, \mathrm{Remove}, \mathrm{Replace}\}, \\ \dfrac{M_t^{\mathrm{base}}}{M_t}, & t \in \{\mathrm{Pitch}, \mathrm{Volume}\}, \end{cases} \tag{8}$$

such that $r_t > 1$ indicates improvement over the baseline. The overall score is computed from the equal-weight geometric mean across all six tasks:

$$\mathrm{TBG}_{\mathrm{all}} = 100 \left[\left(\prod_{t=1}^{6} r_t \right)^{1/6} - 1 \right]. \tag{9}$$

The geometric mean prevents gains on a subset of tasks from overly masking degradation on others. Throughout this work, we use SteerSound-Base as the baseline, Task-Aware CLAP as the primary metric for audio generation, source addition, source removal, and source replacement, Pitch MAE for source pitch modification, and Volume MAE for source volume modification.

Selective Task-Balanced Gain. To more directly assess the model's relative selective editing capability, we report a separate gain score across source removal, source replacement, pitch modification, and volume modification. Let $\mathcal{T}_{\mathrm{sel}} = \{\mathrm{Remove}, \mathrm{Replace}, \mathrm{Pitch}, \mathrm{Volume}\}$. Using the same baseline and task-level ratios r_t defined above, we compute

$$\mathrm{TBG}_{\mathrm{sel}} = 100 \left[\left(\prod_{t \in \mathcal{T}_{\mathrm{sel}}} r_t \right)^{1/|\mathcal{T}_{\mathrm{sel}}|} - 1 \right]. \tag{10}$$

Like $\mathrm{TBG}_{\mathrm{all}}$, this score is zero for SteerSound-Base; positive values indicate aggregate improvement over the baseline.

D Selective Audio Editing Benchmark: Construction and Evaluation

D.1 Benchmark Construction

As discussed in Section 5.1, source overlap in natural acoustic scenes makes it difficult to automatically determine whether the intended source has been correctly edited while non-target content is preserved. We therefore construct the *Selective Audio Editing Benchmark* with controlled source composition and develop a corresponding evaluation protocol for identifying editing failures and selective failure types.

The benchmark is designed to enable large-scale automatic evaluation of both editing success and selective editing failure types under a controlled setting. For audio generation, we follow the same construction strategy as the Complex Audio Editing Data. For source addition, we restrict the reference audio to a single source. For selective editing operations, the reference audio always contains exactly two temporally non-overlapping sources. One serves as the target source to be edited, while the other serves as the non-target source to be preserved. This simplified and controlled source composition facilitates the evaluation of audio generation and source addition while allowing the editing behaviors of the target and non-target sources in selective editing operations to be measured independently.

We reuse the Class-Label Pool $\mathcal{L}$ and Fine-Grained Description Pool $\mathcal{P}$ introduced in Appendix C.1. We first construct task-specific source pools with different audio durations and then sample the required sources according to each editing task to form the reference audio R, target audio T, and editing instruction I. The overall construction procedure is illustrated in Fig. 18.

In addition to the benchmark, we construct training and validation sets from the same controlled distribution to support controlled comparisons among different post-training strategies without introducing distribution differences. The training set contains 10,240 samples per task, resulting in 61,440 samples across the six tasks. The validation set and the benchmark each contain 1,000 samples per task. Throughout the construction process, we use the same audio source pools and the same source-level training/validation/test partition as the Complex Audio Editing data in Appendix C.2, ensuring that audio sources used for validation and benchmark evaluation do not appear in training.

D.2 Automatic Evaluation Protocol

The controlled source composition of the benchmark enables automatic evaluation of both editing success and selective editing failure types.

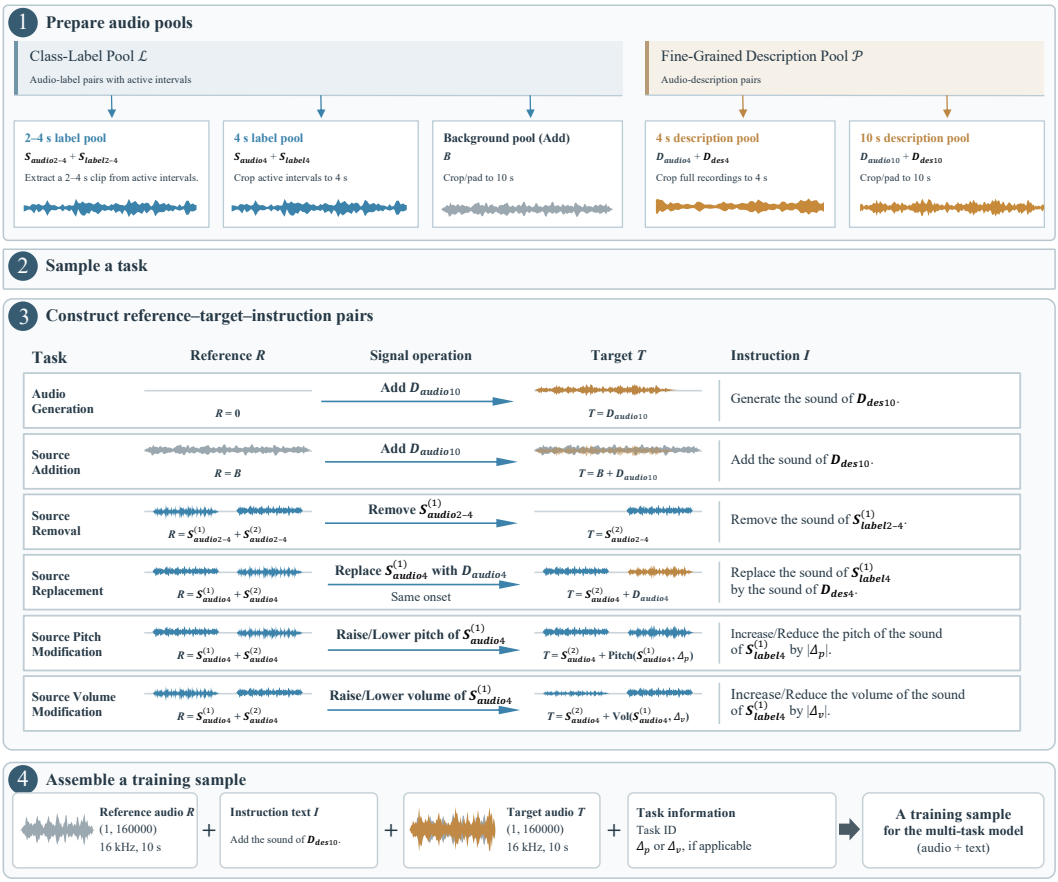

Figure 18: Overview of the Selective Audio Editing Benchmark constructor. Audio sources from the Class-Label Pool $\mathcal{L}$ and Fine-Grained Description Pool $\mathcal{P}$ are organized into task-specific pools and composed into reference–target–instruction pairs for six audio editing operations with controlled source layouts. $|\Delta_p|$ is fixed at 4 semitones, and $|\Delta_v|$ is fixed at 5 dB in this process.

Task Success Identification and Selective Failure Categorization. Depending on the editing operation, we use CLAP similarity to measure source semantics, local RMS-level changes to measure removal and volume modification, and multi-resolution CQT alignment to measure relative pitch changes. The success criterion and failure categorization for each task are defined as follows:

- **Audio Generation.** We compute the CLAP similarity between the generated audio and the target label. The generation is considered successful when the CLAP similarity is greater than 0.08; otherwise, the sample is categorized as *Other*.
- **Source Addition.** We compute the CLAP similarity between the edited audio and the label of the source to be added. The addition is considered successful when the CLAP similarity is greater than 0.04; otherwise, the sample is categorized as *Other*. The preservation of the original source is not included in the success criterion for source addition, as such failures are rare and additional preservation constraints can introduce false negatives.
- **Source Removal.** For source removal, the reference audio contains a source to be removed and a temporally non-overlapping source to be preserved. We measure the local volume changes of the two sources before and after editing. The edit is considered successful when the source to be removed decreases by at least 6 dB, while the absolute volume change of the source to be preserved remains below 3 dB. This design is also motivated by our observation that cases where the source to be preserved remains perceptually present but undergoes substantial waveform-level changes are rare. If neither source is removed, the

Table 7: Automatic success criteria and selective failure categorization for the six editing tasks.

Task	Outcome	Target / edited source	Source to preserve
Audio Generation	**Success**	Target present (CLAP > 0.08)	–
	Other	Target absent (CLAP ≤ 0.08)	–
Source Addition	**Success**	Added source present (CLAP > 0.04)	–
	Other	Added source absent (CLAP ≤ 0.04)	–
Source Removal	**Success**	Removed (≥ 6 dB decrease)	Preserved ($\lvert\Delta v\rvert < 3$ dB)
	Target Missed	Not removed (< 6 dB decrease)	Preserved ($\lvert\Delta v\rvert < 3$ dB)
	Wrong-Target Edit	Not removed (< 6 dB decrease)	Removed (≥ 6 dB decrease)
	Unintended Non-Target Edit	Removed (≥ 6 dB decrease)	Removed (≥ 6 dB decrease)
	Other	Remaining cases	
Source Replacement	**Success**	Replacement source	Original preserved source
	Target Missed	Original target source	Original preserved source
	Wrong-Target Edit	Original target source	Replacement source
	Unintended Non-Target Edit	Replacement source	Replacement source
	Other	Ambiguous or unmatched cases	
Source Pitch Modification	**Success**	Modified ($r \in [0.8, 1.2]$)	Preserved ($\lvert\Delta p\rvert < 1$ semitone)
	Target Missed	Unchanged ($\lvert\Delta p\rvert < 1$ semitone)	Unchanged ($\lvert\Delta p\rvert < 1$ semitone)
	Wrong-Target Edit	Unchanged ($\lvert\Delta p\rvert < 1$ semitone)	Modified ($r \in [0.8, 1.2]$)
	Unintended Non-Target Edit	Modified ($r \in [0.8, 1.2]$)	Modified ($r \in [0.8, 1.2]$)
	Other	Remaining measurable cases	
Source Volume Modification	**Success**	Modified ($r \in [0.8, 1.2]$)	Preserved ($\lvert\Delta v\rvert < 1.5$ dB)
	Target Missed	Unchanged ($\lvert\Delta v\rvert < 1.5$ dB)	Unchanged ($\lvert\Delta v\rvert < 1.5$ dB)
	Wrong-Target Edit	Unchanged ($\lvert\Delta v\rvert < 1.5$ dB)	Modified ($r \in [0.8, 1.2]$)
	Unintended Non-Target Edit	Modified ($r \in [0.8, 1.2]$)	Modified ($r \in [0.8, 1.2]$)
	Other	Remaining cases	

For source replacement, a category is assigned only when its local CLAP similarity exceeds 0.08; if both candidate categories pass the threshold, the one with higher similarity is selected. For pitch and volume modification, Δp and Δv denote the measured pitch and volume changes, respectively, while r denotes the measured signed change divided by the requested signed change. Samples for which the pitch change cannot be measured for one or both sources are excluded from the final statistics.

result is categorized as *Target Missed*; if the source to be preserved is removed instead of the target source, it is categorized as *Wrong-Target Edit*; and if both sources are removed, it is categorized as *Unintended Non-Target Edit*. All remaining failures are categorized as *Other*.

- **Source Replacement.** For source replacement, the reference audio contains a source to be replaced and a temporally non-overlapping source to be preserved. We separately measure the local CLAP similarities within the two source regions. For the source to be replaced, we compare its original label with the replacement-source label; for the source to be preserved, we compare its original label with the replacement-source label. A source category is considered present only when its CLAP similarity is greater than 0.08, and the category with the higher similarity is selected when both satisfy the threshold. The edit is considered successful when the source to be replaced is identified as the replacement source while the source to be preserved retains its original identity. If both sources retain their original identities, the result is categorized as *Target Missed*; if the source to be replaced remains unchanged while the source to be preserved is identified as the replacement source, it is categorized as *Wrong-Target Edit*; and if both regions are identified as the replacement source, it is categorized as *Unintended Non-Target Edit*. All ambiguous or otherwise unmatched cases are categorized as *Other*.
- **Source Pitch Modification.** For source pitch modification, the reference audio contains a source to be modified and a temporally non-overlapping source to be preserved. We separately estimate the relative pitch changes within the two source regions using multi-resolution CQT alignment. For the source to be modified, we compare its measured signed pitch change with the requested change; for the source to be preserved, we measure its pitch change relative to the original source. A source is considered successfully modified when its measured change reaches 80%–120% of the requested signed change, and is considered unchanged when its absolute pitch change is below 1 semitone. The edit is considered successful when the source to be modified is successfully modified while the source to be preserved remains unchanged. If both sources remain unchanged, the result is categorized as *Target Missed*; if the source to be modified remains unchanged while the source to be preserved undergoes the requested change, it is categorized as *Wrong-Target Edit*; and if both sources undergo the requested change, it is categorized as *Unintended Non-Target Edit*. All remaining failures are categorized as *Other*. Samples for which the pitch change cannot be measured for either source are excluded.

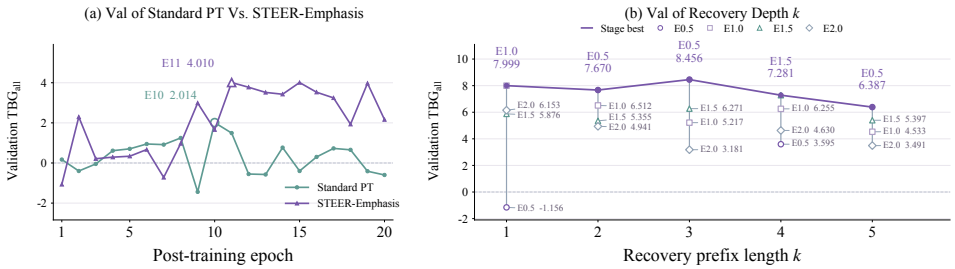

Figure 19: Validation dynamics on the Complex Audio Editing Validation Set. (a) Validation $\mathrm{TBG}_{\mathrm{all}}$ of Standard PT and STEER-Emphasis across post-training epochs. (b) Validation $\mathrm{TBG}_{\mathrm{all}}$ of STEER-Recovery across recovery depths k. Markers denote intermediate checkpoints evaluated every 0.5 epoch at each recovery depth, and the purple curve connects the validation-best checkpoint for each k.

- **Source Volume Modification.** For source volume modification, the reference audio contains a source to be modified and a temporally non-overlapping source to be preserved. We separately measure the local volume changes within the two source regions. For the source to be modified, we compare its measured signed volume change with the requested change; for the source to be preserved, we measure its volume change relative to the original source. A source is considered successfully modified when its measured change reaches 80%–120% of the requested signed change, and is considered unchanged when its absolute volume change is below 1.5 dB. The edit is considered successful when the source to be modified is successfully modified while the source to be preserved remains unchanged. If both sources remain unchanged, the result is categorized as *Target Missed*; if the source to be modified remains unchanged while the source to be preserved undergoes the requested change, it is categorized as *Wrong-Target Edit*; and if both sources undergo the requested change, it is categorized as *Unintended Non-Target Edit*. All remaining failures are categorized as *Other*.

The criteria are summarized in Table 7. The task-level success rate and overall success rate are computed over valid samples.

E Details on the Effectiveness and Robustness of STEER

This appendix complements Section 5.3 with experimental details, validation dynamics, and additional results. Section E.1 describes the training configurations and learning dynamics of STEER on the complex and selective audio editing datasets. Sections E.2 and E.3 provide additional results for the comparison with standard post training and the alternative early-trajectory range, respectively. Finally, Section E.4 compares STEER-Recovery with continued STEER-Emphasis training under matched updates.

E.1 Details on Effectiveness of STEER

Training configuration on the complex dataset. On the Complex Audio Editing Training Set, STEER-Emphasis is initialized from SteerSound-Base and trained for 20 epochs with a global batch size of 480. We use a peak learning rate of 2×10^{-6}, with 1,000 optimizer updates for warmup followed by cosine decay. Checkpoints are selected by validation $\mathrm{TBG}_{\mathrm{all}}$, computed across all six tasks.

STEER-Recovery starts from the STEER-Emphasis checkpoint with the highest validation $\mathrm{TBG}_{\mathrm{all}}$. We train for 2 epochs at each recovery depth k with a global batch size of 480. The checkpoint initialization and optimizer restart procedure follow the selective setting. We use a peak learning rate of 1×10^{-5}, with 64 optimizer updates for warmup followed by a constant learning rate, and retain $P = 2$. Because the complex training set is substantially larger, we evaluate on the validation set every 0.5 epoch.

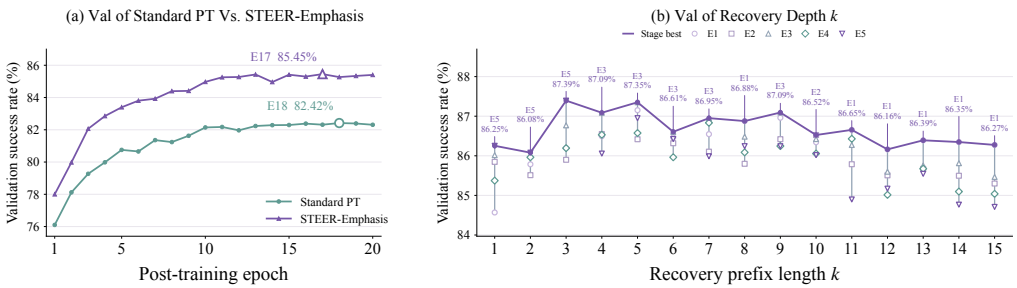

Figure 20: Validation dynamics on the Selective Audio Editing Benchmark. (a) Overall validation success rate of Standard PT and STEER-Emphasis across post-training epochs. (b) Overall validation success rate of STEER-Recovery across recovery depths k. Markers denote checkpoints from the five training epochs at each recovery depth, and the purple curve connects the validation-best checkpoint for each k.

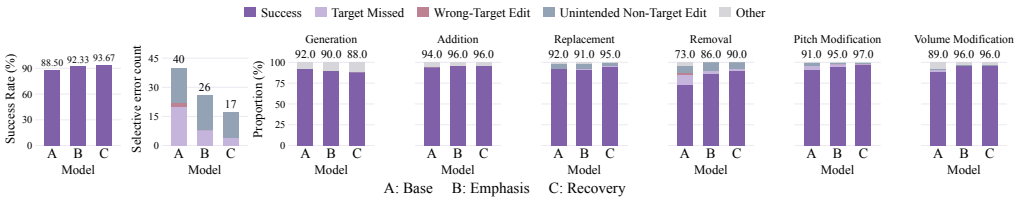

Figure 21: Human-annotated overall and task-wise success rates and selective error distributions of SteerSound-Base, STEER-Emphasis, and STEER-Recovery on the Complex Audio Editing Test Set, using 100 randomly sampled examples per task.

Training configuration on the selective dataset. On the selective audio editing training set, STEER-Emphasis is initialized from SteerSound-Base and trained for 20 epochs on the selective audio editing training set with a global batch size of 480 and a gradient accumulation factor of 1. We use a peak learning rate of 2×10^{-6}, with 1,000 optimizer updates for warmup followed by cosine decay. Checkpoints are selected by the overall validation success rate.

STEER-Recovery is initialized from the validation-best STEER-Emphasis checkpoint. At each recovery depth k, we train for 5 epochs with a global batch size of 480. Each depth starts from the best checkpoint found so far, with the optimizer reinitialized and the learning-rate schedule restarted. We use a peak learning rate of 1×10^{-5}, with 64 optimizer updates for warmup followed by a constant learning rate. The shorter warmup accommodates the limited training duration at each depth. We set the patience parameter to $P = 2$, stopping when two consecutive recovery depths fail to improve the best validation score and retaining the best checkpoint observed so far.

Training dynamics and additional results. Figures 19 (purple curves) and 20 show the validation performance of STEER-Emphasis across training epochs and STEER-Recovery across recovery depths on the two datasets, respectively. Recovery training further improves upon the validation-best STEER-Emphasis checkpoint in both settings, and both datasets achieve their best validation performance at $k = 3$.

On the selective dataset, we additionally continue recovery training beyond the stopping point prescribed by $P = 2$ to examine whether deeper recovery training yields further gains. None of the subsequently evaluated depths surpasses the earlier validation best, supporting the use of the patience-based stopping rule in this setting.

Figure 21 shows the human-annotated task-wise success rates and error-type distributions corresponding to Figure 8 A1. Figure 22 provides the task-level success rates and failure-type distributions corresponding to Figure 9 A1, while Table 2 A1 reports the corresponding metric results on the Complex test set. Taken together, these complementary evaluations show that STEER-Emphasis improves editing success and reduces selective editing failures across both data settings,

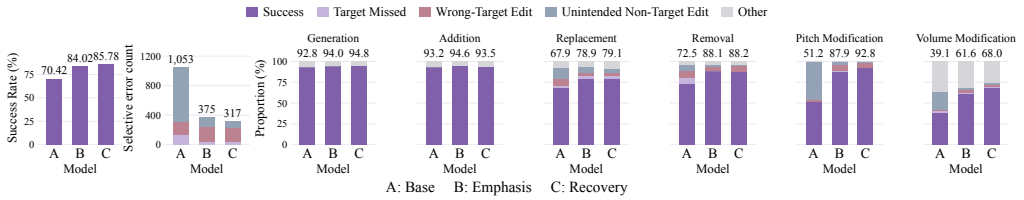

Figure 22: Overall and task-wise success rates and selective error distributions of SteerSound-Base, STEER-Emphasis, and STEER-Recovery on the Selective Audio Editing Benchmark.



Figure 23: Human-annotated overall and task-wise success rates and selective error distributions of Standard PT and STEER-Emphasis on the Complex Audio Editing Test Set, using 100 randomly sampled examples per task.

while STEER-Recovery provides further gains. These results further support the effectiveness of STEER in improving both overall editing performance and source-selection reliability.

E.2 Details on STEER-Emphasis Ablation with Standard PT

For each dataset, Standard PT follows the same training configuration and checkpoint-selection criterion as STEER-Emphasis in Section E.1, differing only in the sampling distribution of the training noise level σ. Standard PT uses logit-normal sampling for all tasks, whereas STEER-Emphasis follows the task-conditioned distribution in Equation 3. Both optimize the same Flow Matching objective.

Figures 19 and 20 compare the validation curves of Standard PT (green) and STEER-Emphasis (purple). STEER-Emphasis maintains higher validation success rates throughout post-training on the selective dataset and consistently higher validation $\mathrm{TBG}_{\mathrm{all}}$ during the later stages of training on the complex dataset. The advantage of early-trajectory emphasis is therefore not confined to the validation-selected checkpoint.

Figure 23 shows the human-annotated task-wise success rates and error-type distributions corresponding to Figure 8 A2. Figure 24 provides the task-wise success rates and failure-type distributions corresponding to Figure 9 A2, while Table 2 A2 reports the corresponding metric results on the Complex test set. Taken together, these evaluations show that STEER-Emphasis achieves higher editing success and reduces more selective editing errors across both data settings compared with Standard PT. These results further support the benefit of emphasizing critical early-trajectory states beyond standard post-training.

E.3 Details on Range Robustness of STEER-Emphasis

STEER-Emphasis$_{15}$ follows the same dataset-specific training configurations and checkpoint-selection criteria as STEER-Emphasis in Section E.1, differing only in noise-level sampling. It uses the shared early-trajectory range identified by the elbow-based rule in Range Robustness of STEER-Emphasis, sampling σ uniformly from $[\sigma_{15}, 1]$ for all selective tasks. Generate and Add retain logit-normal sampling.

Figure 25 provides the task-level comparison corresponding to Figure 9 A3, while Table 2 A3 reports the corresponding complex test results. The shared-range and task-group-specific variants



Figure 24: Overall and task-wise success rates and selective error distributions of Standard PT and STEER-Emphasis on the Selective Audio Editing Benchmark.

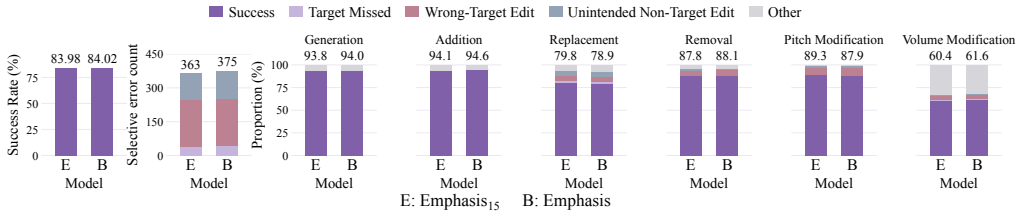

Figure 25: Overall and task-wise success rates and selective error distributions of STEER-Emphasis$_{15}$ and STEER-Emphasis on the Selective Audio Editing Benchmark.

achieve comparable overall performance in both settings. This indicates that the effectiveness of STEER-Emphasis is not tied to the original threshold-based range definition.

E.4 Details on STEER-Recovery Ablation with Matched Updates

To examine whether additional parameter updates alone can account for the gains of STEER-Recovery, we compare it with continued STEER-Emphasis training under matched updates. For each dataset, both runs start from the same validation-best STEER-Emphasis checkpoint. The continuation baseline, denoted STEER-Emphasis$_{\text{cont}}$, retains the STEER-Emphasis noise-level sampling distribution and the standard Flow Matching objective instead of introducing recovery supervision on self-generated trajectory states.

The continuation baseline uses the same global batch size of 480, peak learning rate of 1×10^{-5}, and warmup length of 64 optimizer updates as STEER-Recovery. Unlike recovery training, it uses a single optimizer throughout the run, with one warmup phase followed by cosine decay. We use this uninterrupted schedule to avoid imposing recovery-specific optimization restarts on the continuation baseline. All remaining settings, including checkpoint selection, follow the corresponding dataset-specific configurations in Appendix E.1.

On the selective training set, we continue STEER-Emphasis for 15 epochs, matching the total number of updates across three recovery depths with 5 epochs per depth, and evaluate once per epoch, following the STEER-Recovery setting. On the complex training set, we continue training for 6 epochs, matching three recovery depths with 2 epochs per depth, and evaluate every 0.5 epoch, again following the STEER-Recovery setting. In both settings, the continuation uses the same total number of optimizer updates as those used to obtain the final STEER-Recovery checkpoint at $k = 3$.

Here, *matched updates* refers to the total number of optimizer updates performed through completion of the recovery depth that yields the final checkpoint, not the number of updates incorporated into that checkpoint. STEER-Recovery may revert to an earlier validation-best checkpoint before proceeding to the next depth; updates discarded by this rollback are still included in the total. In contrast, STEER-Emphasis$_{\text{cont}}$ trains continuously without rollback. Its updates therefore accumulate along a single optimization path, which can contain more updates than the path leading to the selected recovery checkpoint. Updates at later recovery depths used for stopping or additional analysis are not included.

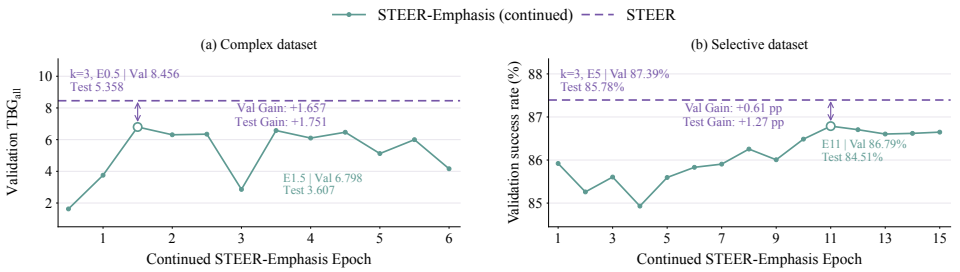

Figure 26: Validation trajectories of continued STEER-Emphasis under matched updates on the Complex and Selective Audio Editing datasets. Dashed lines indicate the validation performance of the selected STEER-Recovery checkpoints.

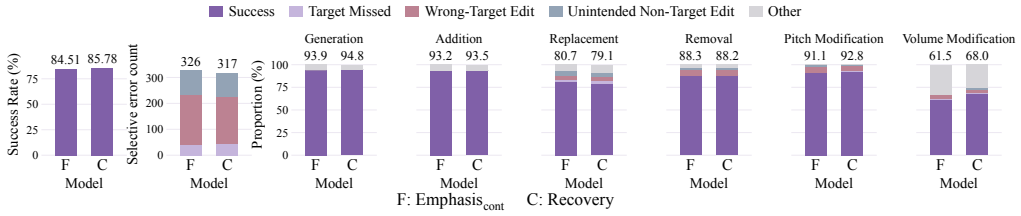

Figure 27: Overall and task-wise success rates and selective error distributions of STEER-Emphasis$_{\text{cont}}$ and STEER-Recovery on the Selective Audio Editing Benchmark.

Figure 26 shows the validation trajectories for the complex and selective settings. Figure 27 provides the task-level breakdown corresponding to Figure 9 A4 . Together with the complex test results in Table 2 A4 , these comparisons show that STEER-Recovery achieves better overall editing performance than the continuation baseline in both settings. These results indicate that the gains of STEER-Recovery cannot be explained solely by additional optimization updates, but instead benefit from learning recovery from self-generated trajectory states.

E.5 Details on Recovery after Early Corruption

Figure 28 provides the task-wise results corresponding to Figure 12. Across all four selective editing tasks, STEER-Emphasis improves robustness to early-trajectory corruption. STEER-Recovery further increases k_{50} substantially for removal, replacement, and pitch modification, while maintaining the gain on volume modification. These results further confirm that STEER-Recovery improves recovery from early-trajectory errors across selective editing tasks.

F Details on Generalization of STEER

To complement the generalization experiments in Section 5.4, this appendix is organized into two parts. Appendix F.1 provides additional experimental details and extended results for STEER on UNISON in unified audio editing. Appendix F.2 provides the corresponding experimental details and extended results for STEER on SAM Audio in audio source removal. Since SAM Audio adopts a distinct timestep formulation, Appendix F.2 additionally describes the adaptation of STEER to its timestep setting.

F.1 Details on the STEER Generalization to UNISON

In this section, we provide supplementary details and results for the generalization experiments of STEER on UNISON. We first describe the training configurations of Standard PT and STEER-Emphasis, followed by those of STEER-Recovery and STEER-Emphasis$_{\text{cont}}$. We then report the training dynamics together with more comprehensive experimental results.

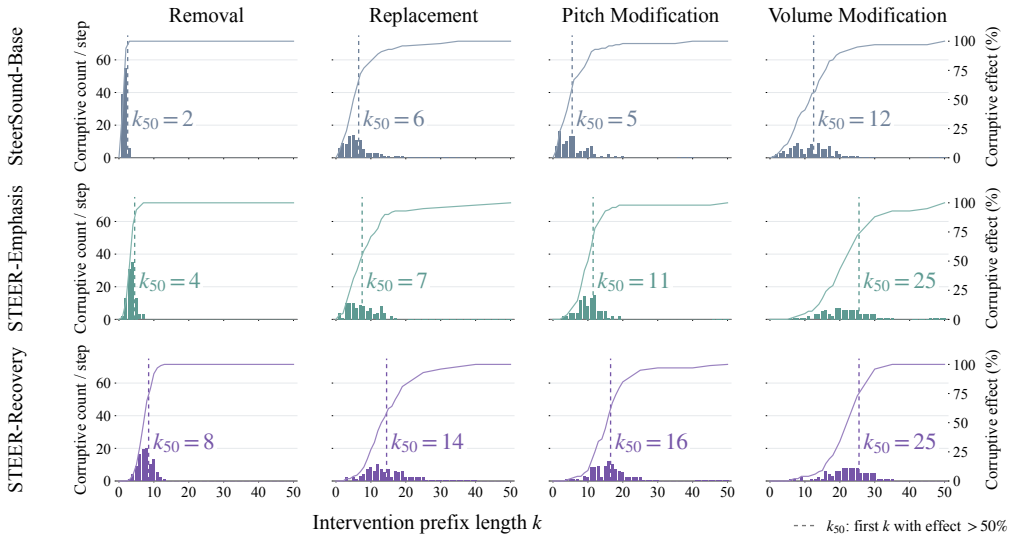

Figure 28: Task-wise recovery from early-trajectory corruption across model variants on the Selective Audio Editing Benchmark. Larger k_{50} indicates stronger robustness to early-trajectory errors.

STEER-Emphasis and Standard PT Starting from the publicly released UNISON Li et al. (2026b) checkpoint, we fully train the DiT backbone while freezing the VAE and Qwen language encoder. Both STEER-Emphasis and Standard PT are trained for 30 epochs on the selective audio editing datasets with a global batch size of 32. We use AdamW with a weight decay of 0.01, gradient clipping at 1.0, and an initial learning rate of 1×10^{-5} with cosine decay and no warmup. Since UNISON supports only generation, addition, removal, and replacement tasks, we restrict both training and evaluation to these four tasks, using the corresponding task subsets from the selective audio editing training datasets for post-training and evaluating on the corresponding test sets. Notably, for STEER-Emphasis on UNISON, we use *the same critical early-trajectory ranges identified for SteerSound-Base*, allowing us to evaluate the generalizability of both STEER-Emphasis and the corresponding early-range definition across model architectures.

STEER-Recovery STEER-Recovery is initialized from the validation-best UNISON checkpoint obtained after STEER-Emphasis. We continue to fully train the DiT backbone while keeping the remaining components frozen. For each recovery depth k, we train for 5 epochs with a batch size of 32 using AdamW, a weight decay of 0.01, and gradient clipping at 1.0. The optimizer is reinitialized at every k. We use 64 optimizer-step linear warmup to a peak learning rate of 1×10^{-5}, followed by a constant learning rate. For UNISON, we likewise set the patience hyperparameter to $P = 2$.

STEER-Emphasis$_{\text{cont}}$. STEER-Emphasis$_{\text{cont}}$ starts from the same validation-best STEER-Emphasis checkpoint as STEER-Recovery and uses the same peak learning rate and batch size. Following the continuation baseline on SteerSound-Base in Appendix E.4, it uses a single optimizer throughout training, with one warmup phase followed by cosine decay, rather than restarting the optimizer and learning-rate schedule at each recovery depth. All remaining training settings are kept consistent with STEER-Recovery. We continue STEER-Emphasis for 20 epochs, matching the total number of updates through the selected STEER-Recovery checkpoint at $k = 4$ (four recovery depths with 5 epochs per depth), and evaluate the model once per epoch.

Training Dynamics and Detailed Results. Figure 29 shows the validation performance of STEER-Emphasis and Standard PT across training epochs, together with the validation results of STEER-Recovery across recovery depths. Throughout training, STEER-Emphasis consistently achieves a higher overall task success rate on the validation set than Standard PT. STEER-Recovery further improves upon the validation-best STEER-Emphasis checkpoint, with the best validation performance obtained at recovery depth $k = 4$. We additionally continue recovery training beyond the stopping point implied by the patience criterion to examine the effectiveness of the stopping

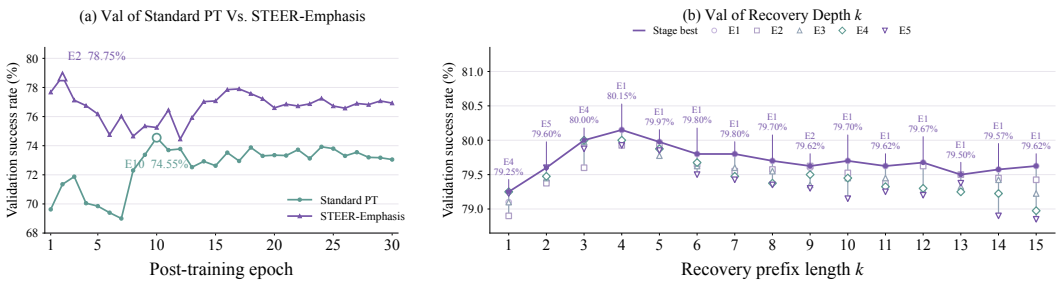

Figure 29: Validation dynamics of STEER on UNISON. (a) Overall validation success rate of Standard PT and STEER-Emphasis across post-training epochs. (b) Overall validation success rate of STEER-Recovery across recovery depths k. Markers denote checkpoints from the five training epochs at each recovery depth, and the purple curve connects the validation-best checkpoint for each k.

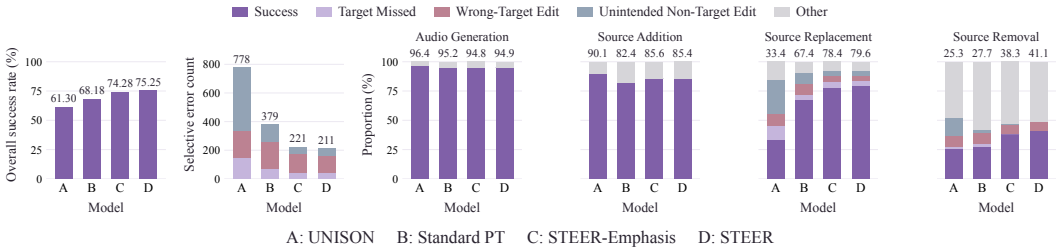

Figure 30: Overall and task-wise success rates and selective error distributions of UNISON under different training strategies.

strategy and the generalizability of $P = 2$. The UNISON results show the same pattern: once no new validation best is achieved for two consecutive recovery depths, deeper recovery training does not surpass the previous best, further supporting the effectiveness of the patience-based stopping criterion.

In addition to the overall success rate shown in Figure 10, we report the task-wise success rates of the final UNISON models in Figure 30. STEER-Emphasis achieves better overall editing performance than Standard PT, with particularly large improvements on selective editing tasks. STEER-Recovery further improves performance on selective editing tasks and achieves additional gains in overall editing performance over STEER-Emphasis$_{\text{cont}}$, as shown in Figures 31 and 32. These results demonstrate the cross-model effectiveness of STEER and further show that the identified critical early-trajectory ranges remain robust across different model architectures.

F.2 Details on the STEER Generalization to SAM Audio

Since SAM Audio adopts a different timestep formulation, we first describe how STEER is adapted to SAM Audio. Specifically, SAM Audio uses the opposite time parameterization from SteerSound: $t = 0$ corresponds to noise, while $t = 1$ corresponds to clean audio. We therefore map the training strategy from the σ parameterization to SAM Audio through

$$t = 1 - \sigma.$$

For STEER-Emphasis, the critical early sampling range for the Remove task, $\sigma \in [\sigma_2, 1]$, is correspondingly mapped to $t \in [0, 1-\sigma_2]$. This preserves the same training semantics of emphasizing the high-noise, early-denoising region. For STEER-Recovery, we retain the same 50-step noise-position grid and convert each position according to $t_j = 1 - \sigma_j$. Starting from the initial noise, we perform a k-step explicit Euler rollout using the conditional velocity field of the model:

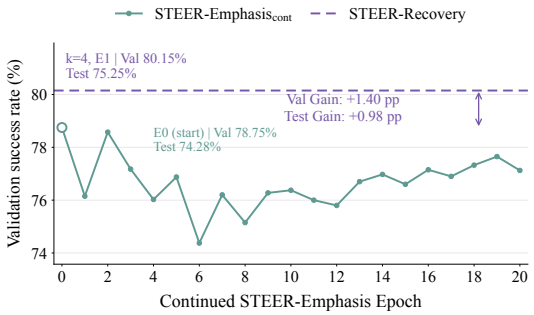

Figure 31: Validation trajectory of continued STEER-Emphasis under matched updates on the Selective Audio Editing Benchmark using UNISON. The dashed line indicates the validation performance of the selected STEER-Recovery checkpoint at $k = 4$.

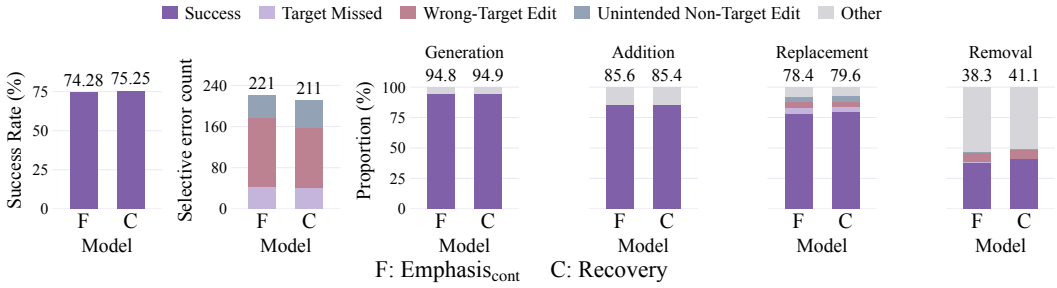

Figure 32: Task-wise comparison between STEER-Emphasis$_{\text{cont}}$ and the selected STEER-Recovery checkpoint at $k = 4$ under matched updates on the Selective Audio Editing Benchmark using UNISON.

$$\mathbf{z}_{j+1} = \mathbf{z}_j + (t_{j+1} - t_j)\mathbf{v}_\theta(\mathbf{z}_j, t_j, c).$$

After reaching the rollout state $\mathbf{z}_k$, we construct the corrective velocity toward the clean target latent as

$$\mathbf{v}_k^* = \frac{\mathbf{z}_{\text{GT}} - \mathbf{z}_k}{1 - t_k},$$

where $\mathbf{z}_{\text{GT}}$ denotes the clean latent obtained by concatenating the encoded target and residual tracks. The Euler rollout is used only to implement the STEER-Recovery post-training procedure and is not used for validation or inference. For all validation and test evaluations involving SAM Audio, we follow its default midpoint ODE solver and integrate the trajectory using `torchdiffeq.odeint` with the default SAM Audio integration step size $\Delta t = \frac{2}{32} = 0.0625$. We also follow the default SAM Audio setting and do not use CFG during inference.

In addition, SAM Audio is a separation model that jointly outputs the target source and the residual audio. We post-train it only on the Remove task, treating the source specified for removal in the editing instruction as the separation target and using the predicted residual audio as the final Remove output for evaluation. During training, both the target and residual branches are jointly supervised.

In the remainder of this section, we provide supplementary details and results for the generalization experiments of STEER on SAM Audio. We first describe the training configurations of STEER-Emphasis and Standard PT, followed by those of STEER-Recovery. We then report the training dynamics together with more comprehensive experimental results.

STEER-Emphasis and Standard PT. We perform post-training based on the publicly released SAM Audio Small Shi et al. (2025) model from Meta. We train only the DiT module, while keeping the audio codec, T5 text encoder, and all other components frozen. For all training settings, we train

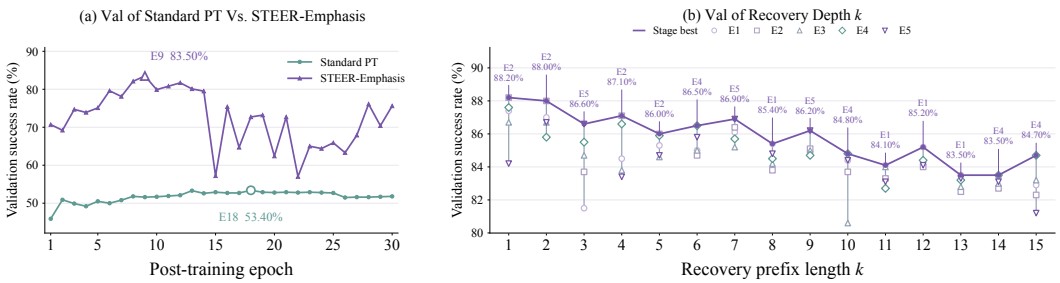

Figure 33: Validation dynamics of STEER on SAM Audio. (a) Validation removal success rate of Standard PT and STEER-Emphasis across post-training epochs. (b) Validation removal success rate of STEER-Recovery across recovery depths k. Markers denote checkpoints from the five training epochs at each recovery depth, and the purple curve connects the validation-best checkpoint for each k.

for 30 full epochs using AdamW with a constant learning rate of 1×10^{-4}, no warmup, no learning-rate scheduler, and a global batch size of 160. As described in Section 5.4, we train SAM Audio only on the removal task and use the same critical early-trajectory range defined for SteerSound-Base, mapped to SAM Audio through $t = 1 - \sigma$.

STEER-Recovery. STEER-Recovery is initialized from the validation-best STEER-Emphasis checkpoint. We continue to train only the DiT module, while keeping the audio codec, T5 text encoder, and all other components frozen. We do not use early stopping and instead train recovery depths from $k = 1$ to $k = 15$, with 5 epochs at each k. We use a global batch size of 768 and AdamW as the optimizer, reinitializing the optimizer at every recovery depth. The learning rate is kept constant at 1×10^{-4} throughout training, with no warmup or learning-rate decay. We also use the same σ positions associated with each recovery depth k as STEER-Recovery on SteerSound-Base, mapped to SAM Audio through $t = 1 - \sigma$.

STEER-Emphasis$_{\text{cont}}$. STEER-Emphasis$_{\text{cont}}$ starts from the same validation-best STEER-Emphasis checkpoint as STEER-Recovery and uses the same peak learning rate and batch size. Following the continuation baseline on SteerSound-Base in Appendix E.4, it uses a single optimizer throughout training, with one warmup phase followed by cosine decay, rather than restarting the optimizer and learning-rate schedule at each recovery depth. All remaining training settings are kept consistent with STEER-Recovery. We continue STEER-Emphasis for 5 epochs, matching the total number of updates through the selected STEER-Recovery checkpoint at $k = 1$ (one recovery depth with 5 epochs), and evaluate the model once per epoch.

Training Dynamics and Detailed Results. Figure 33 shows the validation performance of STEER-Emphasis and Standard PT across training epochs, together with the validation results of STEER-Recovery across recovery depths. Throughout training, STEER-Emphasis consistently achieves a higher overall task success rate on the validation set than Standard PT. STEER-Recovery further improves upon the validation-best STEER-Emphasis checkpoint, with the best validation performance obtained at recovery depth $k = 1$. Although we continue recovery training to deeper k for analysis, no subsequent recovery depth surpasses the previous validation best after two consecutive depths fail to establish a new best. This observation is consistent with the results on SteerSound and UNISON, further supporting the effectiveness and cross-model generalizability of the patience-based stopping criterion with $P = 2$.

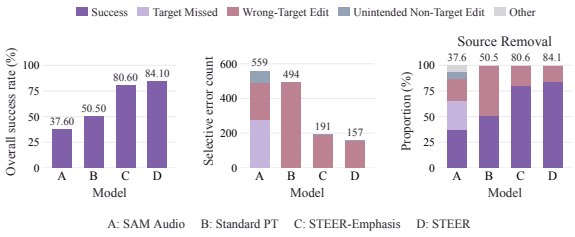

Figure 34: Overall success rates and selective error distributions of SAM Audio under different training strategies.

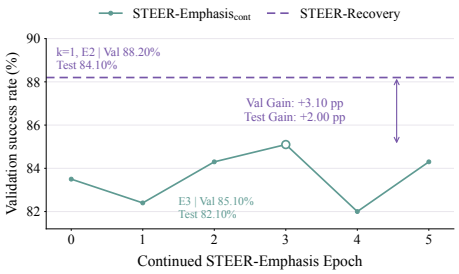

Figure 35: Matched-update validation trajectory on SAM Audio. The dashed line marks the selected STEER-Recovery checkpoint at $k = 1$.

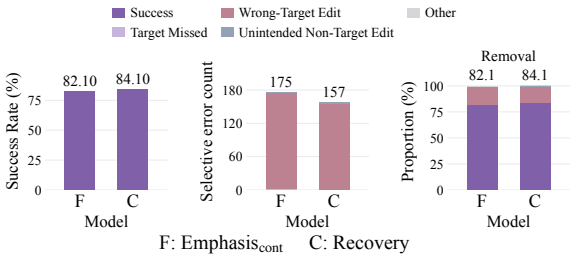

Figure 36: Matched-update comparison of STEER-Emphasis$_{\text{cont}}$ and STEER-Recovery ($k = 1$) on SAM Audio on the Selective Audio Editing Benchmark.

In addition to the removal success rates reported in Figure 10, we further report the number and distribution of selective editing failures under different post-training strategies in Figure 34. STEER-Emphasis substantially improves removal performance over Standard PT while markedly reducing selective editing failures. STEER-Recovery further outperforms STEER-Emphasis$_{\text{cont}}$, achieving a higher removal success rate while further reducing selective editing failures, as shown in Figures 35 and 36.

These results demonstrate the cross-model effectiveness of STEER and the robustness of the identified critical early-trajectory range across model architectures. Moreover, the consistent improvement on SAM Audio, which uses a different ODE solver from SteerSound, demonstrates that STEER remains effective across different inference solvers.

G Details on Transferability to Related Selective-Control Tasks

To complement the experiments in Section 5.5, this appendix is organized into two parts. Appendix G.1 provides additional experimental details and extended results for applying the Early-Trajectory Principle to the fine-grained selective-control task of target speaker extraction (TSE) with SteerSound-Base, while Appendix G.2 provides the corresponding results with SAM Audio. Since TSE differs substantially from conventional label-level audio editing or extraction tasks, each section first introduces the task setting, model adaptation, training configuration, and evaluation metric, followed by the corresponding performance results and analysis.

G.1 Details on Target Speaker Removal with SteerSound-Base

In this section, we first introduce the TSE-like task setting for SteerSound-Base, followed by the model adaptation and training configuration. Finally, we analyze the corresponding training results.

TSE-like Task Setting on SteerSound-Base. Existing audio editing models and tasks often specify the editing target at the category level, e.g., "Remove the sound of [class label]" from a mixture containing sources from different categories. This is also the setting adopted by SteerSound-Base and SAM Audio, where the target source is specified by a class label. In many practical scenarios, however, the target must be identified from substantially finer-grained conditions. Target speaker extraction (TSE) is one such example, where the model may need to distinguish between two same-gender speakers according to the linguistic content spoken by the target speaker. We use this setting to investigate whether the Early-Trajectory Principle also benefits fine-grained selective control.

To further study this question, we extend SteerSound-Base to a TSE-like task in which the model identifies and removes a specified speaker according to speech content. Since SteerSound-Base natively supports source removal rather than source extraction, we formulate the task as *Target Speaker Removal (TSR)*. Given a speech mixture

$$x = s_{\text{target}} + s_{\text{residual}}, \tag{11}$$

where s_{target} denotes the target speaker to be removed and s_{residual} denotes the remaining speech that should be preserved, the model learns

$$\hat{s}_{\text{residual}} = f_\theta(x, c), \tag{12}$$

rather than generating s_{target}. The condition c is constructed from the transcription of the target speaker. During training, the instruction takes the form

$$\{\texttt{Mix audio}\}\ \texttt{Remove the sound of}\ \{\texttt{transcription}\}, \tag{13}$$

where $\{\texttt{Mix audio}\}$ denotes the *input mixed audio*, while $\{\texttt{transcription}\}$ denotes the *text transcription* of the target sound to be removed. For example, if the target speaker says "Please close the door," the corresponding instruction is "{Mix audio} Remove the sound of Please close the door." The model must therefore use the textual content to locate the corresponding speaker within the mixture, remove the sound of that speaker, and preserve the remaining speech.

Model Adaptation, Training Strategy, and Trainable Parameters.

Since this task requires the model to understand the linguistic content of the input audio, we replace the original T5 text encoder in SteerSound-Base with a multimodal large language model (MLLM) and connect the MLLM to the SteerSound DiT through a Meta Query Pan et al. (2025) mechanism. Specifically, we use MOSS-Audio-4B-Thinking Yang et al. (2026) as the audio-content understanding module. MOSS jointly receives the mixed audio, the text removal instruction, and 50 learnable queries inserted after the <SEG> token. We extract the corresponding query hidden states before the output head of MOSS and pass them through a one-layer Transformer Connector followed by a LayerNorm–MLP Projector. The resulting representations are mapped into conditioning features and replace the original T5-based textual conditioning of SteerSound-Base.

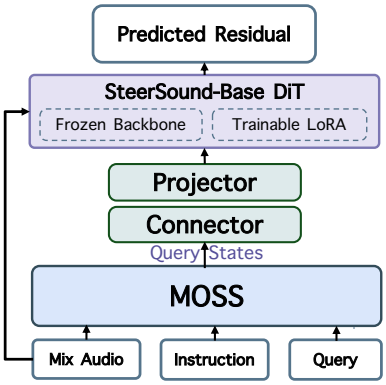

Figure 37: Architecture of the adapted SteerSound-Base model for the TSR task.

This adaptation introduces newly initialized components, including the learnable queries, Transformer Connector, and Projector. Moreover, we observe that standard Flow Matching training barely converges on this fine-grained task and fails to learn effective target selection. Therefore, rather than first training these newly introduced components with standard Flow Matching and subsequently applying STEER-Emphasis as a post-training stage, we directly compare standard training with training that incorporates the early-focused principle from the beginning. This setting allows us to more clearly assess the effect of emphasizing the early trajectory throughout training. The architecture of the adapted SteerSound-Base model for TSR task is shown in Figure 37.

Since fine-grained speaker editing differs substantially from the original audio editing tasks, we adapt the SteerSound-Base DiT with LoRA to avoid disrupting its previously learned editing capability. Specifically, LoRA modules are inserted into the `qkv` and `linear` layers of the attention blocks, with rank 64, alpha 128, and dropout 0.05. We jointly train the DiT LoRA parameters, learnable queries, Transformer Connector, and Projector, while keeping all other components frozen. The numbers of trainable parameters for different modules are summarized in Table 8.

Training Configuration. We initialize the model from SteerSound-Base and train it for 20 epochs using either Standard PT or the early-focused training strategy. Both settings use AdamW with a weight decay of 0.01, a constant learning rate of 1×10^{-4}, no warmup or learning-rate decay, a batch size of 64, and gradient clipping with a maximum norm of 1.0.

Table 8: Trainable parameters.

Module	Params.
DiT LoRA	15.73M
Queries	0.13M
Connector	52.46M
Projector	8.53M
Total	**76.84M**

Since several trainable components, including the DiT LoRA, Connector, and Projector, are newly initialized, this setting does not fully satisfy the original assumption of STEER-Emphasis that the model already possesses an initial editing capability. We there-

fore adopt a mixed sampling strategy: while most training states follow the early-focused σ sampling of STEER-Emphasis, a small proportion are sampled using the standard logit-normal distribution to retain coverage of the full generation trajectory. Specifically, Standard PT uses the standard logit-normal sampling strategy $\sigma \sim \mathrm{sigmoid}\big(\mathcal{N}(0,1)\big)$. For early-focused training, we use

$$\sigma \sim \begin{cases} \mathcal{U}(0.97, 1), & \text{with probability } 0.8, \\ \mathrm{sigmoid}\big(\mathcal{N}(0,1)\big), & \text{with probability } 0.2. \end{cases}$$

Thus, 80% of the training states are concentrated in the high-noise early-trajectory region, while the remaining 20% retain standard logit-normal sampling to preserve coverage of the full generation trajectory. The training objective remains the standard Flow Matching objective in both settings.

Training Dataset. We use the PromptTSE Seki et al. (2025) dataset for training and evaluation. In PromptTSE, each mixed-audio sample contains speech from two distinct speakers and is accompanied by the target-speaker audio to be extracted, the residual speech after removing the target speaker, and the transcription of the target speaker. In our TSR setting, the provided residual audio is used as the generation target, while the target transcription is used to construct the fine-grained removal instruction.

Evaluation Metric. We directly use CLAP Audio–Audio similarity for TSR ($\text{CLAP-AA}_{\text{TSR}}$) to measure the similarity between the generated residual audio and the ground-truth residual audio. Let $E_a(\cdot)$ denote the CLAP Elizalde et al. (2023) audio encoder. For sample i,

$$\text{CLAP-AA}_{\text{TSR}i} = \frac{E_a(\hat{s}_{\text{residual},i})^\top E_a(s_{\text{residual},i})}{\|E_a(\hat{s}_{\text{residual},i})\|_2 \, \|E_a(s_{\text{residual},i})\|_2}.$$

The final score is averaged over all N evaluation samples:

$$\text{CLAP-AA}_{\text{TSR}} = \frac{1}{N} \sum_{i=1}^{N} \text{CLAP-AA}_{\text{TSR}i}\,.$$

$\text{CLAP-AA}_{\text{TSR}}$ therefore directly measures the audio similarity between the generated residual audio and the ground-truth residual audio.

Experimental Results. Figure 38 shows the validation $\text{CLAP-AA}_{\text{TSR}}$ of Standard PT and early trajectory focused training over 20 training epochs. Compared with Standard PT, early trajectory focused training learns substantially more effectively. This result further demonstrates the importance of emphasizing the training on critical early-trajectory states for fine-grained selective control task.

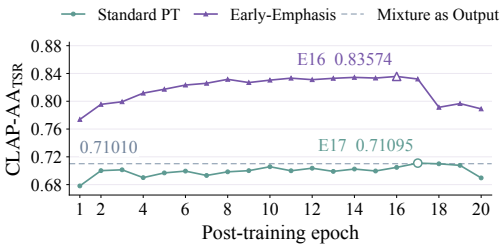

Figure 38: Validation $\text{CLAP-AA}_{\text{TSR}}$ during training on the TSR task with SteerSound-Base. The horizontal line denotes the $\text{CLAP-AA}_{\text{TSR}}$ of the mixture audio.

The horizontal line denotes the $\text{CLAP-AA}_{\text{TSR}}$ obtained by directly using the mixed audio as the model output and comparing it with the ground-truth residual audio. Notably, without emphasizing the training on critical early trajectory, Standard PT performs close to the direct-mixture baseline, whereas emphasizing the training on critical early trajectory substantially improves performance on this fine-grained selective control task.

G.2 Details on Target Speaker Separation with SAM Audio

In this section, we first introduce the TSE-like task setting for SAM Audio, followed by the model adaptation and training configuration. Finally, we analyze the corresponding training results.

TSE-like Task Setting on SAM Audio. Following the SteerSound-Base experiment, we further investigate whether the early-trajectory training principle benefits fine-grained selective control on SAM Audio. Unlike SteerSound-Base, which natively supports source removal, SAM Audio is designed for audio source separation. We therefore directly formulate the task as *Target Speaker Separation (TSS)*. Given a speech mixture

$$x = s_{\text{target}} + s_{\text{residual}}, \tag{14}$$

where s_{target} denotes the target speaker to be separated and s_{residual} denotes the remaining speech, the model learns

$$[\hat{s}_{\text{target}}, \hat{s}_{\text{residual}}] = f_\theta(x, c), \tag{15}$$

with supervision applied to both $\hat{s}_{\text{target}}$ and $\hat{s}_{\text{residual}}$. The condition c is constructed from the transcription of the target speaker. During training, the instruction takes the form

$$\{\texttt{Mix audio}\}\ \texttt{Separate the sound of}\ \{\texttt{transcription}\}. \tag{16}$$

The meanings of the individual components in the instruction are the same as those described in Section G.1.

Model Adaptation, Training Strategy, and Trainable Parameters.

Since this task requires the model to understand the linguistic content of the input audio, we follow the adaptation used for SteerSound-Base and replace the original T5 text encoder in SAM Audio with MOSS-Audio-4B-Thinking Yang et al. (2026). We connect MOSS to the SAM Audio DiT through the same Meta Query Pan et al. (2025) mechanism. The overall architecture is identical to that shown in Figure 37, except that the SteerSound-Base DiT is replaced by the SAM Audio DiT.

Table 9: Trainable parameters.

Module	Params.
SAM Audio LoRA	127.40M
Queries	0.13M
Connector	65.54M
Projector	18.09M
Total	**211.17M**

We similarly adapt the SAM Audio DiT with LoRA, using rank 256, alpha 512, and dropout 0.05. We also directly compare standard Flow Matching training with training that incorporates the early-focused principle from the beginning. During training, we use the same set of trainable components as described in Section G.1. The numbers of trainable parameters for different modules are summarized in Table 9.

Training Configuration. We initialize the model from the publicly released SAM Audio Small Shi et al. (2025) checkpoint and train it on this task for 20 epochs using either Standard PT or the early-focused training strategy. Both settings use AdamW with a weight decay of 0.01, a peak learning rate of 5×10^{-5}, 1,000 optimizer-step warmup followed by cosine decay, a batch size of 56, and gradient clipping with a maximum norm of 1.0.

Since several trainable components, including the DiT LoRA, Connector, and Projector, are newly initialized, this setting does not fully satisfy the original assumption of STEER-Emphasis that the model already possesses an initial task capability. We therefore adopt the same mixed sampling strategy as in the SteerSound-Base experiment: while most training states follow the early-focused t sampling, a small proportion are sampled using the standard logit-normal distribution to retain coverage of the full generation trajectory. Specifically, Standard PT uses the standard logit-normal timestep sampling strategy $t \sim \text{sigmoid}\big(\mathcal{N}(0,1)\big)$. For early-focused training, we use

$$t \sim \begin{cases} \mathcal{U}(0, 0.03), & \text{with probability } 0.8, \\ \text{sigmoid}\big(\mathcal{N}(0,1)\big), & \text{with probability } 0.2. \end{cases}$$

As described in Appendix F.2, SAM Audio adopts the opposite time parameterization from SteerSound, where $t = 0$ corresponds to noise and $t = 1$ corresponds to clean audio. Accordingly, 80% of the training states are concentrated in the high-noise early-trajectory region, while the remaining 20% retain standard logit-normal sampling to preserve coverage of the full generation trajectory. The training objective remains the standard Flow Matching objective in both settings.

Training Dataset. We use the same PromptTSE Seki et al. (2025) dataset as in the SteerSound-Base experiment. In our TSS setting, the target-speaker audio and residual speech are used as the supervision targets for the two output streams.

Evaluation Metric. We use CLAP Audio–Audio similarity for TSS (CLAP-$\mathrm{AA}_{\mathrm{TSS}}$) to evaluate the separation results of SAM Audio. Unlike CLAP-$\mathrm{AA}_{\mathrm{TSR}}$, which only measures the similarity between the generated residual audio and the ground-truth residual audio, CLAP-$\mathrm{AA}_{\mathrm{TSS}}$ jointly evaluates both separated outputs while penalizing similarity to the incorrect source. Let $E_a(\cdot)$ denote the CLAP Elizalde et al. (2023) audio encoder, and define the cosine similarity between two audio signals as

$$C(a,b) = \frac{E_a(a)^\top E_a(b)}{\|E_a(a)\|_2 \, \|E_a(b)\|_2}.$$

For sample i, the model produces the predicted target speech $\hat{s}_{\mathrm{target},i}$ and predicted residual speech $\hat{s}_{\mathrm{residual},i}$, with corresponding ground-truth signals $s_{\mathrm{target},i}$ and $s_{\mathrm{residual},i}$. We define

$$\begin{aligned}\text{CLAP-AA}_{\mathrm{TSS},i} = \frac{1}{2}\Big[&C(\hat{s}_{\mathrm{target},i}, s_{\mathrm{target},i}) - C(\hat{s}_{\mathrm{target},i}, s_{\mathrm{residual},i}) \\ &+ C(\hat{s}_{\mathrm{residual},i}, s_{\mathrm{residual},i}) - C(\hat{s}_{\mathrm{residual},i}, s_{\mathrm{target},i})\Big].\end{aligned}$$

The final score is averaged over all N evaluation samples:

$$\text{CLAP-AA}_{\mathrm{TSS}} = \frac{1}{N}\sum_{i=1}^{N} \text{CLAP-AA}_{\mathrm{TSS},i}\,.$$

CLAP-$\mathrm{AA}_{\mathrm{TSS}}$ therefore encourages the predicted target speech to be more similar to the target ground truth than to the residual speech, while simultaneously encouraging the predicted residual speech to be more similar to the residual ground truth than to the target speech. A higher score indicates better separation of the target and non-target speakers.

Experimental Results.

Figure 39 shows the validation CLAP-$\mathrm{AA}_{\mathrm{TSS}}$ of Standard PT and early trajectory focused training over 20 training epochs. Compared with Standard PT, early trajectory focused training learns this fine-grained speaker separation task substantially more effectively. This result further demonstrates the importance of emphasizing the training on critical early-trajectory states for fine-grained selective control. Together with the SteerSound-Base results, it further shows that this early-focused training principle remains effective across different model architectures and task formulations.

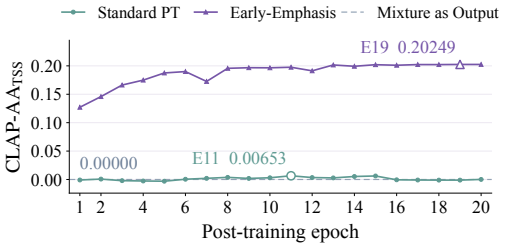

Figure 39: Validation CLAP-$\mathrm{AA}_{\mathrm{TSS}}$ during training on the TSS task with SAM Audio. The horizontal line denotes the CLAP-$\mathrm{AA}_{\mathrm{TSS}}$ of the mixture audio.

The horizontal line denotes the CLAP-$\mathrm{AA}_{\mathrm{TSS}}$ obtained by directly using the unseparated mixed audio as the model output. Standard PT performs close to this direct-mixture baseline, whereas emphasizing the training on the critical early trajectory substantially improves performance on this fine-grained selective control task.

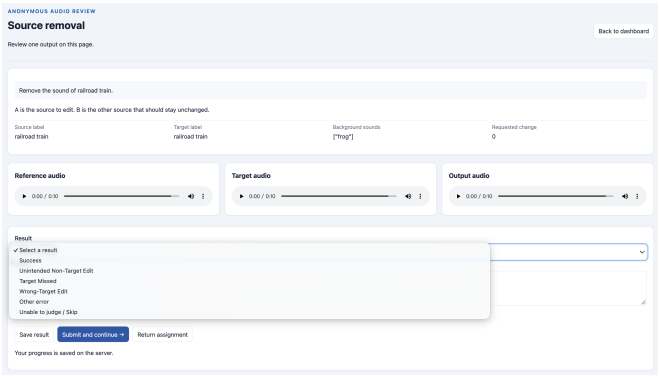

Figure 40: Human evaluation interface for the Complex Audio Editing Test Set.

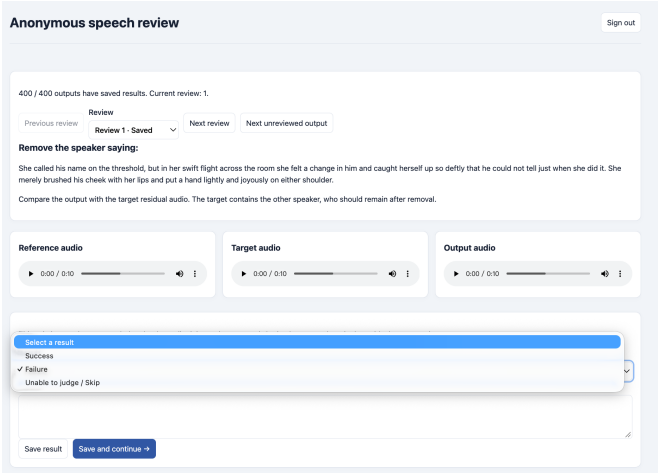

Figure 41: Human evaluation interface for the TSE-related tasks.

H Human Evaluation and Website Design

H.1 Human Annotation Procedure and Criteria

As reported in Sections 5.3 and 5.5, and Appendices E and G, we conduct human evaluations of SteerSound variants on the Complex Audio Editing Test Set and of SteerSound and SAM Audio on the TSE-like tasks.

For the Complex Audio Editing Data, we largely follow the error-type definitions in Appendix D.2. The only difference is that the input audio may contain multiple sources. Therefore, a sample is categorized as Unintended Non-Target Edit whenever the target source undergoes the intended edit while at least one non-target source is also modified. For the TSR and TSS tasks, annotators are asked to determine whether the output audio satisfies the task requirement and to label each result as either successful or unsuccessful. For both annotation tasks, annotators are allowed to abstain when a sample is too complex to judge reliably. Such samples are excluded from the final statistics. During annotation, annotators are not informed which model produced each audio sample.

H.2 Website Design

We develop two web-based annotation interfaces for human evaluation. For the Complex Audio Editing Test Set, annotators are shown the editing instruction and the reference, target, and output audio, and classify each result according to the success and selective-error categories (Figure 40). For the TSE-related tasks, we use a simplified interface in which annotators judge each output as Success, Failure, or Unable to Judge (Figure 41). Model identities are hidden in both interfaces.