ARMONIA: EDIT-COMPATIBLE DIFFUSION INVERSION FOR TEXT-GUIDED MUSIC EDITING

Anonymous authors
Paper under double-blind review

ABSTRACT

Text-guided music editing aims to modify specified attributes of a recording while preserving musical content unrelated to the intended change, enabling flexible music creation. Inversion anchors this editing process to the source music by mapping it to a state that initializes generation under the target text. However, a state that enables faithful source reconstruction is not necessarily a suitable starting point for the requested edit. This distinction motivates us to move part of the editing control into inversion and prepare an initialization that is better suited to the requested edit. To this end, we propose *Armonia*, a training-free method that refines inversion states using output-level attention distillation derived from source- and target-conditioned model responses. To improve refinement efficiency and effectiveness while reducing the risk of off-manifold drift, Armonia uses temporal weighting to focus optimization on selected inversion steps and local projection to limit deviations from each DDIM inversion proposal. The resulting state is then used to initialize the original target-conditioned denoising process, leaving the pretrained diffusion model unchanged. Compared with SOTA, Armonia achieves an average relative improvement of 11.5% in CLAP across three datasets while retaining competitive structural preservation.

1 INTRODUCTION

Text-guided music editing aims to modify specified attributes of a source recording while preserving content unrelated to the requested change. Advances in text-conditioned diffusion models provide a generative foundation for this task by enabling music synthesis from natural-language descriptions Liu et al. (2023; 2024); Ghosal et al. (2023); Evans et al. (2025; 2026); Gong et al. (2025; 2026). Building on this capability, zero-shot music editing methods extend pretrained generators from creating new music to modifying existing recordings without task-specific training.

The key challenge in extending pretrained generators to editing is how to anchor target-conditioned generation to a specific source recording. Inversion-based methods address this by mapping the source into a noisy latent state, from which denoising under the target condition produces the edited result Song et al. (2021); Manor & Michaeli (2024). Recent methods further exploit information obtained during inversion to preserve source content throughout editing Liu et al. (2025); Yang et al. (2026). In the piano-to-violin example in Fig. 1 (right, a), source attention control produces a result with low audible distortion, but the requested timbre change remains incomplete. This motivates a closer examination of whether an initialization constructed for source reconstruction also provides a suitable starting point for the requested target-conditioned edit.

Reconstruction-oriented inversion seeks a latent that allows the model to faithfully reproduce the source recording. Editing requires the model to use this latent under the target condition to realize the requested change while preserving unrelated content. This difference in requirements constitutes a *reconstruction–editing objective gap*. High reconstruction fidelity therefore does not by itself guarantee that the inverted latent is a suitable starting point for the requested edit.

Motivated by this gap, *Armonia* moves part of the editing control into inversion, as illustrated in Fig. 1 (right, b). It uses output-level attention distillation to incorporate target guidance into the evolving latent, preparing an initialization for the requested edit. Temporal weighting focuses refinement on selected steps, while local projection bounds displacement from each inversion proposal. The resulting initialization is then passed directly to native target-conditioned denoising.

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.

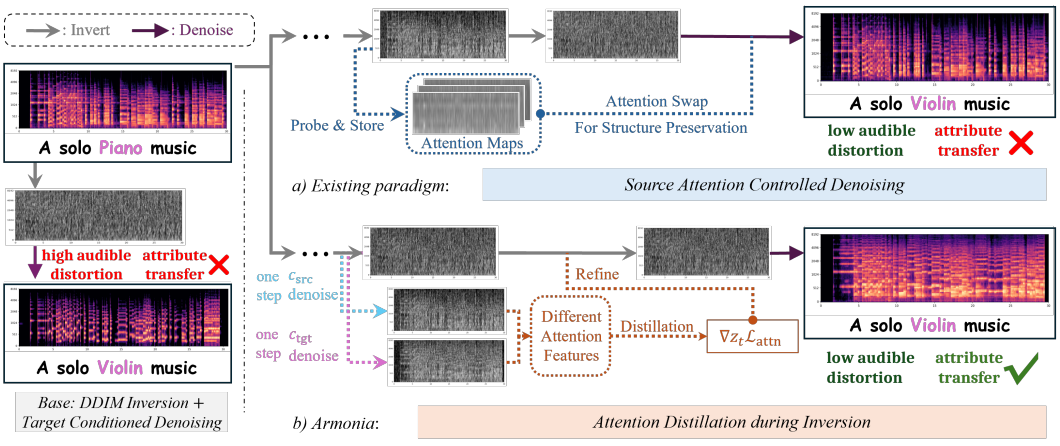

Figure 1: **Illustration of motivation.** Left: DDIM inversion followed by target-conditioned denoising yields audible distortion and incomplete timbre conversion. Right: (a) Source attention control aims to preserve musical content during denoising Yang et al. (2026); Liu et al. (2025), but timbre conversion remains incomplete. (b) Armonia incorporates target guidance through attention distillation during inversion, preparing an initialization for the conversion with low audible distortion.

Extensive experiments on MelodiaEdit Yang et al. (2026), MusicDelta Bittner et al. (2014), and ZoME-Bench Liu et al. (2025) demonstrate that Armonia improves target adherence successfully. Armonia achieves an average relative improvement of 11.5% in CLAP over SOTA while maintaining competitive source structure preservation across the three datasets and the subjective listening evaluations. Results highlight the practical value of incorporating editing guidance into inversion.

Our contributions can be summarized as follows:

1) We identify a *reconstruction–editing objective gap* in inversion-based zero-shot music editing: most inversion techniques are designed to reconstruct the source, while the resulting states are subsequently expected to support a target-conditioned change and preserve unrelated musical content.

2) We propose *Armonia*, a training-free framework that refines inversion states through output-level attention distillation with temporal weighting and local projection, preparing the latent initialization for the requested edit while retaining native target-conditioned denoising.

3) We demonstrate Armonia's superior target adherence through extensive objective and subjective evaluations on MelodiaEdit, MusicDelta, and ZoME-Bench, achieving an average CLAP improvement of 11.5% over SOTA while maintaining competitive structure preservation and music quality.

2 Related Work

2.1 Text-Guided Music Editing

Training-based methods include task-specific training and instruction tuning. AUDIT Wang et al. (2023) and InstructME Han et al. (2023) learn audio and music editing, respectively, from paired editing data and instructions. Instruct-MusicGen adapts pretrained MusicGen Copet et al. (2023) with trainable audio and text conditioning modules for stem-level editing Zhang et al. (2024a). Both routes incur data preparation and training costs.

Training-free methods remove this additional training stage by controlling pretrained models Liu et al. (2023; 2024); Ghosal et al. (2023) at inference time. Inversion carries source information into generation through recovered noise Manor & Michaeli (2024), while attention constraints and feature reuse guide content preservation during target-conditioned denoising Zhang et al. (2024b); Liu et al. (2025). Melodia uses attention probing to select self-attention layers for temporal structure guidance and reuses their features during denoising Yang et al. (2026). Moreover, for polyphonic music, Polyphonia combines acoustic priors with attention calibration to localize timbre edits to

a target stem while preserving non-target stems Li et al. (2026). SteerMusic optimizes source-initialized latents through delta denoising score (DDS) distillation, without diffusion inversion Niu et al. (2026). Together, these studies explore inversion, attention control, and latent optimization for text-guided music editing. Such preservation must be accompanied by sufficient target realization, since retaining source content alone does not establish that the requested edit has been achieved.

2.2 Diffusion Inversion

Diffusion inversion enables editing of observed samples by recovering latent representations for pretrained generators. DDIM inversion approximately reverses deterministic sampling Song et al. (2021). Edit-friendly DDPM inversion recovers a sequence of noise maps for reconstruction and subsequent manipulation Huberman-Spiegelglas et al. (2024). Reconstruction accuracy can be improved through unconditional-embedding optimization Mokady et al. (2023), iterative refinement of inversion estimates Garibi et al. (2024), or correction of source-branch deviations during denoising Ju et al. (2024). Beyond reconstruction accuracy, another line of work examines how the recovered representation affects editability. Source Prompt Disentangled Inversion reduces dependence on the source prompt Li et al. (2024), while Editable Noise Map Inversion improves editability by refining noise representations to reduce discrepancies between reconstructed and edited noise maps Kang & Choi (2025). Related techniques also support audio editing, with DDPM inversion enabling zero-shot text-guided modification Manor & Michaeli (2024) and iterative inversion being adapted to flow-matching music models Lan et al. (2024). Taken together, these studies highlight the need for inverted representations to support both source reconstruction and the intended edit.

3 Preliminaries

Text-conditioned Latent Diffusion Models (LDMs) generate audio through conditional denoising in the latent space of a pretrained autoencoder Ho et al. (2020); Rombach et al. (2022). Given an audio signal x, an encoder $\mathcal{E}$ produces a clean latent $z_0 = \mathcal{E}(x)$. The diffusion process is indexed by $t = 0, \ldots, T$, where $t = 0$ corresponds to the clean latent and T denotes the maximum diffusion timestep. The forward process perturbs z_0 at timestep t as

$$z_t = \sqrt{\bar{\alpha}_t} z_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0, I), \tag{1}$$

where β_t is the noise variance schedule, $\alpha_t = 1 - \beta_t$, and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$ controls the signal retained at timestep t. The denoiser, expressed in noise-prediction form as $\epsilon_\theta(z_t, t, c)$, estimates the noise in z_t given a text condition c. During generation, a sampling rule converts the predicted noise into successive latent updates, and the final latent is decoded into audio.

DDIM inversion approximately reverses deterministic DDIM sampling ($\eta = 0$) Song et al. (2021). We write updates between adjacent timesteps; for subsampled schedules, the same rules apply between consecutive selected noise levels. Given latent z_t and condition c, the estimated clean latent is

$$\tilde{z}_0(z_t, t, c) = \frac{z_t - \sqrt{1 - \bar{\alpha}_t}\, \epsilon_\theta(z_t, t, c)}{\sqrt{\bar{\alpha}_t}}. \tag{2}$$

Deterministic DDIM sampling uses this estimate to move from t to the lower-noise timestep $t - 1$:

$$z_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\, \tilde{z}_0(z_t, t, c) + \sqrt{1 - \bar{\alpha}_{t-1}}\, \epsilon_\theta(z_t, t, c). \tag{3}$$

Exact reversal from t to $t + 1$ would require a noise prediction at the unknown higher-noise state. DDIM inversion instead uses the noise prediction at the available lower-noise state under the source condition c_{src} to approximate this update Li et al. (2024):

$$\hat{z}_{t+1} = \sqrt{\bar{\alpha}_{t+1}}\, \tilde{z}_0(\hat{z}_t, t, c_{\mathrm{src}}) + \sqrt{1 - \bar{\alpha}_{t+1}}\, \epsilon_\theta(\hat{z}_t, t, c_{\mathrm{src}}). \tag{4}$$

Standard DDIM inversion maps $\hat{z}_0 = z_0$ to $\hat{z}_T$ under c_{src}. Target guidance c_{tgt} is applied only during denoising from $\hat{z}_T$ to generate the edited audio.

Self-Attention aggregates the denoiser's hidden features H across positions using query, key, and value projections Vaswani et al. (2017):

$$\begin{aligned} Q = HW_Q, \quad K = HW_K, \quad V = HW_V, \\ O(Q, K, V) = A(Q, K)V = \mathrm{Softmax}\Big(\tfrac{QK^\intercal}{\sqrt{d}}\Big) V. \end{aligned} \tag{5}$$

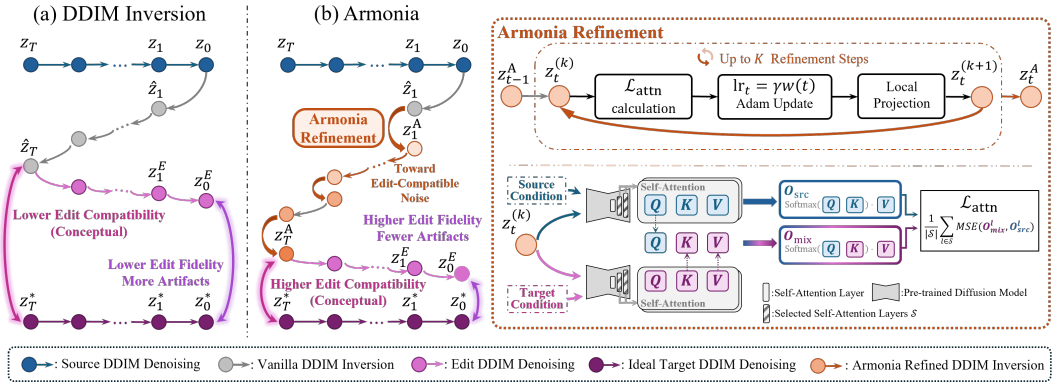

Figure 2: **Overview of Armonia.** Left: (a) Standard DDIM inversion constructs a source-conditioned initialization for target-conditioned denoising. (b) Armonia refines intermediate inversion states while retaining native target-conditioned denoising. Upper right: At each active timestep t, the DDIM proposal from z_{t-1}^{A} initializes $z_t^{(0)}$. Up to K updates refine $z_t^{(k)}$ at fixed t. Temporal weighting selects active timesteps and scales the Adam learning rate; Local Projection bounds the candidate's displacement from the current proposal. The next DDIM proposal is computed from the selected z_t^{A}. Lower right: Each refinement iteration evaluates source and target conditions with the same frozen model at the same $(z_t^{(k)}, t)$. Source queries and detached target keys and values form the mixed attention output O_{mix}. The loss $\mathcal{L}_{\text{attn}}$ averages the mean squared error between O_{mix} and the source output O_{src} over selected self-attention layers $\mathcal{S}$. See Algorithm 1 for the full procedure.

where W_Q, W_K, and W_V are projection matrices and d is the query/key dimension. Queries Q and keys K determine attention weights, while values V provide the aggregated content.

4 Method

Let $z_0 = \mathcal{E}(x_{\text{src}})$ encode the source recording x_{src}, and let c_{src} and c_{tgt} denote the source-side inversion condition and target text condition, respectively. To bridge the reconstruction–editing objective gap, we consider two questions: ***(1)*** *How can source and target conditions define a refinement criterion for the current inversion state without a paired edited reference?* ***(2)*** *How can this criterion guide inversion toward the requested edit while limiting changes to unrelated source content?*

As shown in Fig. 2, *Armonia* refines intermediate states during source-conditioned inversion to prepare an initialization for the requested edit. The resulting state initializes native target-conditioned denoising. All pretrained model parameters remain fixed.

Edit Guidance via Attention Distillation (Sect. 4.1) addresses the first question by constructing $\mathcal{L}_{\text{attn}}$ from source- and target-conditioned responses at the same candidate latent and timestep. **Efficient and Controlled Inversion Refinement** (Sect. 4.2) addresses the second through temporal weighting and Local Projection, which control refinement computation and latent displacement.

4.1 Edit Guidance via Attention Distillation

The first question requires a refinement criterion derived from the source and target conditions alone. Fig. 3 contrasts a piano source with a violin rendering: the edit must preserve the phrase while changing its timbre. DDIM leaves fragmented harmonic traces, while ENM Kang & Choi (2025), which matches one-step predictions under both conditions, changes the timbre but introduces abrupt spectral and envelope changes within notes. Because state matching does not distinguish such disruptions from intended acoustic changes, we compare conditional features after attention aggregation.

Both conditions are evaluated at the same candidate $z = z_t^{(k)}$ and timestep t.

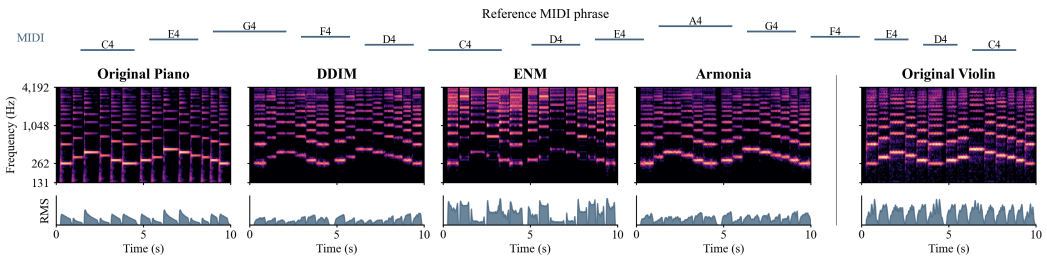

Figure 3: **Design Rationale of Edit Guidance.** The top row sketches the reference MIDI phrase. The five columns show the piano source, DDIM, ENM, and Armonia outputs, and a violin rendering of the same phrase. The piano and violin share the same note sequence and timing but differ in harmonic energy distribution and envelope shape. Edited outputs also differ in harmonic continuity and temporal energy patterns. Motivated by these observations, Armonia measures refinement discrepancies in conditional attention features rather than one-step denoising states.

Using O from Eq. 5, we construct the source and mixed outputs for each selected self-attention (SA) layer ℓ:

$$O^{\ell}_{\mathrm{src}} = O\left(Q^{\ell}_{\mathrm{src}}, K^{\ell}_{\mathrm{src}}, V^{\ell}_{\mathrm{src}}\right), \tag{6}$$

$$O^{\ell}_{\mathrm{mix}} = O\left(Q^{\ell}_{\mathrm{src}}, K^{\ell}_{\mathrm{tgt}}, V^{\ell}_{\mathrm{tgt}}\right), \tag{7}$$

Both outputs share the source queries; subscripts denote the condition used in each evaluation.

The source and mixed outputs define the Output-Level Attention Distillation Loss:

$$\mathcal{L}_{\mathrm{attn}}(z,t) = \frac{1}{|\mathcal{S}|}\sum_{\ell\in\mathcal{S}} \mathrm{MSE}\left(O^{\ell}_{\mathrm{mix}}, O^{\ell}_{\mathrm{src}}\right), \tag{8}$$

where $\mathcal{S}$ denotes the set of selected SA layers. The criterion is evaluated at the current candidate under both conditions and requires no paired edited recording or latent.

Why the output-level criterion guides the edit. The target keys and values are detached (Appendix D), so $\nabla_z \mathcal{L}_{\mathrm{attn}}$ reaches z only through the source-conditioned branch. Matching outputs rather than maps matters because identical maps yield different outputs when values differ. Melodia matches SA maps Yang et al. (2026) and MelodyEdit has target queries read source keys and values Liu et al. (2025); our source queries instead read target keys and values. Descent guides the edit provided the gap between the two responses is dominated by the condition rather than by z. The assumption is not automatic: the source queries can flatten into a uniform map, or the source branch can be driven off its own reconstruction. The constraints of Sect. 4.2 exclude both regimes, and Sect. 5.3 supports this account.

In Fig. 3, Armonia changes the acoustic pattern while retaining the pitch progression and continuity within notes. The remaining question is how to allocate refinement across inversion timesteps and control latent deviation.

4.2 Efficient and Controlled Inversion Refinement

Armonia uses the gradient of $\mathcal{L}_{\mathrm{attn}}$ to refine intermediate states during source-conditioned inversion, starting from the current DDIM proposal b_t with $z_t^{(0)} = b_t$. This guidance adds computation and can push the source branch off its reconstruction, so Temporal Weighting bounds which noise levels are refined and Local Projection bounds the resulting displacement.

Temporal Weighting. Gradient sensitivity varies across noise levels along unmodified DDIM inversion trajectories (Figure 5). This motivates a nonuniform refinement schedule. Temporal Weighting uses a Gaussian window to modulate the learning rate:

$$w(t) = \exp\left[-\frac{(t-\mu)^2}{2\sigma^2}\right], \qquad \mathrm{lr}_t = \gamma w(t), \tag{9}$$

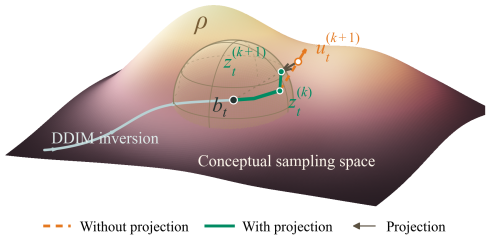

Figure 4: **Local Projection.** The ball bounds drift from b_t with budget ρ. The brown arrow projects an unconstrained update back to its boundary. The geometry is schematic.

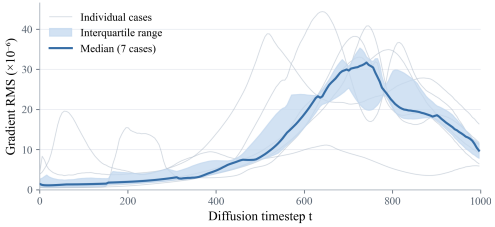

Figure 5: **Temporal sensitivity.** Gradient RMS of $\mathcal{L}_{\text{attn}}$ along DDIM inversion for seven instrument edits. The dark curve and shading denote the median and interquartile range.

where μ, σ, and γ are the window center, width, and base learning rate. Refinement is skipped when $w(t) < \tau$, where τ is the activation threshold. At an active timestep, Adam produces an unconstrained candidate:

$$u_t^{(k+1)} = \text{AdamStep}\Big(z_t^{(k)}, \nabla_z \mathcal{L}_{\text{attn}}(z_t^{(k)}, t); \text{lr}_t\Big). \tag{10}$$

Local Projection. Temporal Weighting controls update strength but does not bound cumulative latent displacement. Large departures from the source-conditioned proposal may alter musical content beyond the requested edit. A lower attention loss alone does not rule out such changes. Local Projection therefore keeps refinement within the neighborhood of b_t shown in Figure 4:

$$\begin{aligned} \mathcal{C}_t &= \left\{ z : \frac{1}{D}\|z - b_t\|_2^2 \le \rho \right\}, \\ z_t^{(k+1)} &= \Pi_{\mathcal{C}_t}\left(u_t^{(k+1)}\right). \end{aligned} \tag{11}$$

Here D is the number of latent elements, ρ the mean-squared displacement budget, and $\Pi_{\mathcal{C}_t}$ the Euclidean projection onto $\mathcal{C}_t$. After up to K inner updates a candidate is accepted as z_t^{A}; when refinement is skipped, $z_t^{\text{A}} = b_t$. The accepted state produces the next DDIM proposal, and $z_{T_{\text{start}}}^{\text{A}}$ initializes target-conditioned denoising.

5 Experiments

5.1 Experimental Setups

Datasets. **MelodiaEdit** Yang et al. (2026) comprises 2,015 music–prompt pairs for instrument, style, and mood editing. These include 1,090 pairs with AudioLDM 2-generated music and 925 pairs based on recordings from **Medley-solos-DB** Lostanlen & Cella (2016) and **GTZAN** Tzanetakis & Cook (2002). **MusicDelta** is a subset of **MedleyDB** Bittner et al. (2014), with editing annotations provided by **MedleyMDPrompts** Manor & Michaeli (2024). The released annotations contain 107 source descriptions and 696 target prompts for 34 recordings. **ZoME-Bench** Liu et al. (2025) contains 1,100 samples drawn from **MusicCaps** Agostinelli et al. (2023) across ten editing categories. Our evaluation focuses on its instrument, genre, and mood subsets.

Baselines. We compare Armonia with eight zero-shot baselines in three groups. (1) **Inversion**: DDIM Inversion Song et al. (2021), DDPM-Friendly Manor & Michaeli (2024), and Null-text Inversion (NTI) Mokady et al. (2023). (2) **DDIM with attention control**: Melodia Yang et al. (2026) and MusicMagus Zhang et al. (2024b). (3) **Editing without inversion**: SDEdit Meng et al. (2021), MusicGen Copet et al. (2023), and SteerMusic Niu et al. (2026). Each baseline uses the hyperparameters recommended by its original paper or released with its official implementation. Method details and configurations are provided in Appendix C.

Table 1: Objective and subjective evaluation on MelodiaEdit, MusicDelta, and ZoME-Bench. First, Second, and Third indicate the three best reported values for each metric within each dataset.

	Method	Objective Evaluation					Subjective Evaluation		
		CLAP ↑	CQT1-PCC ↑	LPAPS ↓	Chroma ↑	KAD ↓	REL ↑	CON ↑	OMQ ↑
MelodiaEdit	DDIM Inversion	0.355	0.263	4.574	0.652	0.096	3.043	3.138	3.091
	DDPM-Friendly	0.340	0.326	4.068	0.703	0.071	2.845	3.540	3.192
	NTI	0.355	0.048	5.200	0.518	0.073	3.038	2.239	2.640
	Melodia	0.373	0.235	3.164	0.681	0.051	3.272	3.635	3.953
	MusicMagus	0.273	0.381	3.356	0.732	0.084	1.670	3.937	2.966
	SDEdit	0.336	0.084	5.459	0.488	0.154	2.793	2.124	2.460
	MusicGen	0.363	0.001	5.113	0.587	0.154	3.139	2.389	2.765
	SteerMusic	0.361	0.262	3.026	0.694	0.074	3.015	3.665	3.512
	Armonia	0.430	0.226	3.658	0.662	0.052	4.124	3.407	3.704
MusicDelta	DDIM Inversion	0.300	0.176	5.930	0.261	0.211	3.210	2.781	2.997
	DDPM-Friendly	0.321	0.222	5.661	0.275	0.202	3.395	2.984	3.190
	NTI	0.251	0.158	5.862	0.254	0.173	2.759	2.738	2.749
	Melodia	0.342	0.483	4.008	0.323	0.141	3.592	3.881	3.595
	MusicMagus	0.198	0.278	5.059	0.288	0.183	2.288	3.285	2.785
	SDEdit	0.167	0.024	6.822	0.201	0.355	2.192	2.029	2.015
	MusicGen	0.224	-0.002	6.271	0.236	0.242	2.517	2.343	2.431
	SteerMusic	0.319	0.335	4.346	0.300	0.225	3.365	3.598	3.476
	Armonia	0.387	0.313	4.787	0.287	0.167	4.271	3.390	3.796
ZoME-Bench	DDIM Inversion	0.235	0.269	5.822	0.248	0.161	2.229	3.086	3.158
	DDPM-Friendly	0.235	0.294	5.700	0.255	0.149	2.231	3.204	3.218
	NTI	0.215	0.310	4.859	0.268	0.153	2.016	3.532	3.274
	Melodia	0.288	0.376	3.899	0.286	0.120	2.410	3.911	3.739
	MusicMagus	0.222	0.322	4.822	0.272	0.175	1.995	3.585	3.340
	SDEdit	0.121	0.051	6.848	0.187	0.283	1.165	2.067	2.033
	MusicGen	0.284	0.007	6.416	0.219	0.279	2.761	2.330	3.048
	SteerMusic	0.268	0.508	3.638	0.304	0.124	2.568	4.306	3.911
	Armonia	0.306	0.350	4.666	0.273	0.144	3.211	3.676	3.904

Objective Metrics. **CLAP** Wu et al. (2023) measures target alignment through cosine similarity between edited-audio and target-text embeddings. Source preservation is evaluated by **LPAPS** (learned perceptual audio patch similarity) Iashin & Rahtu (2021); Manor & Michaeli (2024), **Chroma** Copet et al. (2023) (chroma consistency), and **CQT1-PCC** Niu et al. (2026) (pitch-contour consistency). **Kernel Audio Distance (KAD)** Chung et al. (2025) compares the edited and source audio distributions. Implementation details are in Appendix E.

Subjective Metrics. A five-point listening study with 67 participants covers 30 editing cases (ten per dataset). Separate ratings assess relevance (REL: target alignment), consistency (CON: preservation of source pitch contour and musical structure), and overall musical quality (OMQ: clarity, naturalness, and coherence). See Appendix H for the questionnaire and procedure.

Implementation Details. Armonia uses pretrained AudioLDM 2 Liu et al. (2024) with frozen model weights and a classifier-free guidance (CFG) scale of 3.5. MelodiaEdit uses a 200-step schedule with full inversion; MusicDelta and ZoME-Bench use a 200-step schedule with an inversion ratio of 0.8. Attention-layer selection, refinement hyperparameters, and implementation details are provided in Appendix D.

5.2 Main Evaluation

Objective Evaluation. Table 1 shows that Armonia achieves the highest CLAP on all three datasets. Compared with Melodia, the strongest CLAP baseline, relative gains are 15.2%, 13.1%, and 6.2% on MelodiaEdit, MusicDelta, and ZoME-Bench, respectively, averaging 11.5%. Although Melodia and SteerMusic achieve higher source similarity, Armonia remains competitive, ranking among the top three in LPAPS, CQT1-PCC and KAD on MusicDelta and ZoME-Bench. The spec-

Table 2: Component ablation on MelodiaEdit. AD, TW, and LP denote attention distillation, temporal weighting, and local projection, respectively. Scores are averaged equally across the five subsets.

Components			Evaluation metrics				
AD	TW	LP	CLAP ↑	CQT1-PCC ↑	LPAPS ↓	Chroma ↑	KAD ↓
×	×	×	0.337	0.210	3.761	0.647	**0.042**
✓	×	×	0.374	0.191	4.157	0.629	0.096
✓	✓	×	0.413	0.210	3.847	0.649	0.065
✓	×	✓	0.426	0.222	3.758	0.655	0.058
✓	✓	✓	**0.430**	**0.226**	**3.658**	**0.662**	0.052

Table 3: Guidance-objective comparison on 40 MusicDelta edits from seven source recordings.

Objective	CLAP ↑	CQT1-PCC ↑	LPAPS ↓	Chroma ↑	KAD ↓
DDIM (no refinement)	0.309	0.462	4.684	0.802	0.206
ENM-style	0.311	0.431	4.684	0.784	0.213
Text-CA output	0.337	0.516	**4.288**	0.814	**0.202**
LoA-CA output	0.332	**0.518**	4.407	0.813	0.218
SA map	0.352	0.471	4.493	0.800	0.214
SA output (Armonia objective)	**0.356**	0.494	4.407	**0.816**	0.217

trograms in Figure 7 illustrate this balance, showing spectral changes alongside recognizable source temporal patterns. Listening results are linked in the supplementary materials.

Figure 6 compares four methods on the Generated Instrument subset of MelodiaEdit. Armonia reaches its highest CLAP of 0.458 at full inversion depth, where target adherence and source preservation are best balanced.

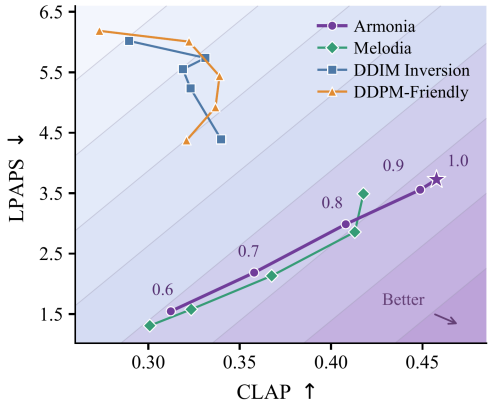

Figure 6: Inversion-depth trade-offs on Generated Instrument. All methods use ratios 0.6–1.0.

Subjective Evaluation. The study further supports Armonia's advantage in target realization. Armonia receives the highest REL on all three datasets, exceeding the strongest baseline by 0.85, 0.68, and 0.45 points on MelodiaEdit, MusicDelta, and ZoME-Bench, respectively. Its OMQ ranks first on MusicDelta and second on MelodiaEdit and ZoME-Bench. Although Melodia and SteerMusic receive higher CON, Armonia surpasses both in REL across all three datasets while maintaining consistently strong musical-quality ratings. More details are provided in Appendix H.

5.3 Ablation Studies and Analysis

Component Ablation. Table 2 compares component configurations on MelodiaEdit under identical sampling settings. Attention distillation alone raises CLAP from 0.337 (DDIM) to 0.374, but worsens source preservation and KAD. This indicates that the loss gradient is informative before any refinement schedule or constraint is applied. Conversely, removing the projection budget from the full configuration lowers CLAP from 0.430 to 0.413, which is consistent with the degenerate decreases discussed in Sect. 4.1 becoming reachable once latent displacement is unbounded. Adding either temporal weighting or local projection improves all five metrics over distillation alone. The full configuration achieves the best CLAP (0.430), CQT1-PCC, LPAPS (3.658), and Chroma. Configurations are detailed in Appendix F.

Choice of Guidance Objective. As shown in Table 3, on 40 MusicDelta edits, Armonia's self-attention output objective achieves the highest CLAP among the tested objectives. Compared with

Table 4: Window shape, position, width, and projection ablations on MelodiaEdit.

Setting				CLAP ↑	CQT1-PCC ↑	LPAPS ↓	Chroma ↑	KAD ↓
Window shape	Position μ	Width σ	ρ					
Uniform	800	120	10^{-6}	0.427	0.215	3.682	0.662	0.058
Gaussian	600	120	10^{-6}	0.411	0.256	3.495	0.674	0.049
Gaussian	1000	120	10^{-6}	0.423	0.227	3.634	0.659	0.051
Gaussian	800	60	10^{-6}	0.419	0.241	3.600	0.667	0.055
Gaussian	800	240	10^{-6}	0.427	0.197	3.784	0.665	0.065
Gaussian	800	120	10^{-7}	0.426	0.224	3.655	0.659	0.049
Gaussian	800	120	10^{-5}	0.425	0.211	3.811	0.653	0.063
Gaussian	800	120	10^{-6}	0.430	0.226	3.658	0.662	0.052

self-attention map matching, it improves CQT1-PCC, LPAPS, and Chroma, with a small increase in KAD. Text-CA output yields the lowest LPAPS and KAD, while LoA-CA output gives the highest CQT1-PCC. Detailed Explantions of different settings are provided in Appendix F.

Temporal Window Design. Table 4 compares window shape, position, and width. Gaussian weighting improves four metrics over uniform weighting at the same active timesteps, with unchanged Chroma. While $\mu = 600$ favors preservation, $\mu = 800$ gives the highest CLAP. Narrowing to $\sigma = 60$ improves CQT1-PCC, LPAPS, and Chroma but lowers CLAP from 0.430 to 0.419; widening to $\sigma = 240$ lowers CLAP and worsens CQT1-PCC, LPAPS, and KAD. The default $(\mu, \sigma) = (800, 120)$ balances alignment and preservation; Appendix F.2 provides further analysis.

Local Projection Budget. The default $\rho = 10^{-6}$ in Table 4 outperforms the no-projection configuration in Table 2 on all five metrics, with CLAP rising from 0.413 to 0.430 and LPAPS falling from 3.847 to 3.658. Tightening the budget to 10^{-7} slightly improves LPAPS and KAD but lowers CLAP and pitch-based similarities. Relaxing it to 10^{-5} worsens all five metrics relative to the default. See Appendix F.3 for the full sweep and selection criterion.

Hyperparameter Analysis. On MusicDelta, an inversion ratio of 0.8 maximizes CLAP, whereas shallower inversion favors preservation. On MelodiaEdit, $K = 3$ and up-block layers 8–9 give the best CLAP among their respective settings. On Generated Instrument, a learning rate of 10^{-3} favors perceptual and pitch-contour preservation; increasing CFG strengthens target alignment at their expense. Appendix G.1 provides the full comparisons.

Efficiency and Memory Usage. With projection, temporal weighting reduces active refinement steps from 200 to 91, with 41.2% lower recorded editing time on three generated MelodiaEdit subsets. In a separate seven-edit, 100-step benchmark, Armonia takes 44.52 seconds per edit: slower than DDIM but faster than NTI, with higher peak memory than both (Appendix G.2).

6 Conclusion

Armonia incorporates editing objectives into inversion, using training-free, output-level attention distillation to prepare latent initializations while retaining the pretrained model and native target-conditioned denoising. Across three evaluated datasets, it achieves the highest target alignment among the compared training-free methods in both objective and subjective evaluations, with strong musical-quality ratings. Ablations show that Armonia can improve alignment and source similarity together. However, source preservation remains weaker than with Melodia and SteerMusic, highlighting the remaining trade-off between target realization and source retention, and it costs more time and computational resources than the standard DDIM Inversion due to editing performance-aware inversion refinement. Future work will explore more selective refinement constraints to better retain musical content unrelated to the requested edit while sustaining target adherence.

AI USE STATEMENT

We used generative AI tools to assist with literature discovery and synthesis, research discussions, method implementation and evaluation scripting, interpretation of experimental results, manuscript organization, English drafting, translation and polishing, and LaTeX editing. The authors reviewed the AI-assisted text, checked references against the cited sources, and verified the code and reported experimental results. We did not use these tools to create new benchmark datasets or to assist with mathematical proofs. The synthetic audio in the existing benchmarks is part of the previously released datasets described in the paper; audio generation and editing by the evaluated models constitute the experimental procedure. The authors take responsibility for the final text, citations, implementation, results, and claims.

REPRODUCIBILITY STATEMENT

Section 4 and Appendices B, C, and D describe the algorithm, the baseline configurations, and the implementation. Appendices E, F, and H document the evaluation metrics, ablation settings, and human listening-study protocol. The supplementary materials provide access to anonymous audio demonstrations, including source audio, source descriptions, target prompts, and edited outputs from Armonia and baseline methods. These curated examples complement the aggregate results reported in the paper.

ETHICS STATEMENT

The listening study involved 67 participants and 30 editing cases. Participants provided informed consent, and the study was covered by institutional ethics approval. Age and musical expertise were collected as participant background information. The paper reports aggregate listening-study results, and the supplementary materials contain no participant records or individual demographic data. The music excerpts shared through the anonymous demonstration page are permitted to be redistributed in this form. Music-editing systems can be misused to create unauthorized derivatives or misleading attributions; their use should respect the rights of composers, performers, and other rightsholders, as well as the terms governing the underlying data and models.

REFERENCES

Andrea Agostinelli, Timo I. Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, Matt Sharifi, Neil Zeghidour, and Christian Frank. Musiclm: Generating music from text. *arXiv preprint arXiv:2301.11325*, 2023.

Rachel M Bittner, Justin Salamon, Mike Tierney, Matthias Mauch, Chris Cannam, and Juan Pablo Bello. Medleydb: A multitrack dataset for annotation-intensive mir research. In *Ismir*, volume 14, pp. 155–160, 2014.

Yoonjin Chung, Pilsun Eu, Junwon Lee, Keunwoo Choi, Juhan Nam, and Ben Sangbae Chon. Kad: No more fad! an effective and efficient evaluation metric for audio generation. *arXiv preprint arXiv:2502.15602*, 2025.

Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez. Simple and controllable music generation. *Advances in neural information processing systems*, 36:47704–47720, 2023.

Zach Evans, Julian D Parker, CJ Carr, Zack Zukowski, Josiah Taylor, and Jordi Pons. Stable audio open. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025.

Zach Evans, Julian D Parker, Matthew Rice, CJ Carr, Zack Zukowski, Josiah Taylor, and Jordi Pons. Stable audio 3. *arXiv preprint arXiv:2605.17991*, 2026.

Daniel Garibi, Or Patashnik, Andrey Voynov, Hadar Averbuch-Elor, and Daniel Cohen-Or. Renoise: Real image inversion through iterative noising. In *European Conference on Computer Vision*, pp. 395–413. Springer, 2024.

Deepanway Ghosal, Navonil Majumder, Ambuj Mehrish, and Soujanya Poria. Text-to-audio generation using instruction guided latent diffusion model. In *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 3590–3598, 2023.

Junmin Gong, Sean Zhao, Sen Wang, Shengyuan Xu, and Joe Guo. Ace-step: A step towards music generation foundation model. *arXiv preprint arXiv:2506.00045*, 2025.

Junmin Gong, Yulin Song, Wenxiao Zhao, Sen Wang, Shengyuan Xu, Jing Guo, and Xuerui Yang. Ace-step 1.5: Pushing the boundaries of open-source music generation. *arXiv preprint arXiv:2602.00744*, 2026.

Bing Han, Junyu Dai, Weituo Hao, Xinyan He, Dong Guo, Jitong Chen, Yuxuan Wang, Yanmin Qian, and Xuchen Song. Instructme: An instruction guided music edit and remix framework with latent diffusion models. *arXiv preprint arXiv:2308.14360*, 2023.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Inbar Huberman-Spiegelglas, Vladimir Kulikov, and Tomer Michaeli. An edit friendly ddpm noise space: Inversion and manipulations. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12469–12478. IEEE, 2024.

Vladimir Iashin and Esa Rahtu. Taming visually guided sound generation. *arXiv preprint arXiv:2110.08791*, 2021.

Xuan Ju, Ailing Zeng, Yuxuan Bian, Shaoteng Liu, and Qiang Xu. Direct inversion: Boosting diffusion-based editing with 3 lines of code. In *The Twelfth International Conference on Learning Representations*, 2024.

Mingyu Kang and Yong Suk Choi. Editable noise map inversion: Encoding target-image into noise for high-fidelity image manipulation. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 28904–28920, 2025.

Gael Le Lan, Bowen Shi, Zhaoheng Ni, Sidd Srinivasan, Anurag Kumar, Brian Ellis, David Kant, Varun Nagaraja, Ernie Chang, Wei-Ning Hsu, et al. High fidelity text-guided music editing via single-stage flow matching. *arXiv preprint arXiv:2407.03648*, 2024.

Haowen Li, Tianxiang Li, Yi Yang, Boyu Cao, and Qi Liu. Polyphonia: Zero-shot timbre transfer in polyphonic music with acoustic-informed attention calibration. *arXiv preprint arXiv:2605.10203*, 2026.

Ruibin Li, Ruihuang Li, Song Guo, and Lei Zhang. Source prompt disentangled inversion for boosting image editability with diffusion models. In *European Conference on Computer Vision*, pp. 404–421. Springer, 2024.

Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark Plumbley. Audioldm: Text-to-audio generation with latent diffusion models. In *Proceedings of the 40th International Conference on Machine Learning, PMLR 2023*, volume 202, pp. 21450–21474. International Machine Learning Society (IMLS), 2023.

Haohe Liu, Yi Yuan, Xubo Liu, Xinhao Mei, Qiuqiang Kong, Qiao Tian, Yuping Wang, Wenwu Wang, Yuxuan Wang, and Mark D Plumbley. Audioldm 2: Learning holistic audio generation with self-supervised pretraining. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024.

Huadai Liu, Jialei Wang, Xiangtai Li, Wen Wang, Qian Chen, Rongjie Huang, Yang Liu, Jiayang Xu, Zhou Zhao, and Wei Xue. Melodyedit: Zero-shot music editing with disentangled inversion control. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 10083–10092, 2025.

Vincent Lostanlen and Carmine-Emanuele Cella. Deep convolutional networks on the pitch spiral for musical instrument recognition. *arXiv preprint arXiv:1605.06644*, 2016.

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.

Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6038–6047. IEEE, 2023.

Xinlei Niu, Kin Wai Cheuk, Jing Zhang, Naoki Murata, Chieh-Hsin Lai, Michele Mancusi, Woosung Choi, Giorgio Fabbro, Wei-Hsiang Liao, Charles Patrick Martin, et al. Steermusic: Enhanced musical consistency for zero-shot text-guided and personalized music editing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 2000–2010, 2026.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

George Tzanetakis and Perry Cook. Musical genre classification of audio signals. *IEEE Transactions on speech and audio processing*, 10(5):293–302, 2002.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

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.

Yusong Wu, Ke Chen, Tianyu Zhang, Yuchen Hui, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

Yi Yang, Haowen Li, Tianxiang Li, Boyu Cao, Xiaohan Zhang, Liqun Chen, and Qi Liu. Melodia: Training-free music editing guided by attention probing in diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 2209–2217, 2026.

Yixiao Zhang, Yukara Ikemiya, Woosung Choi, Naoki Murata, Marco A Martínez-Ramírez, Liwei Lin, Gus Xia, Wei-Hsiang Liao, Yuki Mitsufuji, and Simon Dixon. Instruct-musicgen: Unlocking text-to-music editing for music language models via instruction tuning. *arXiv preprint arXiv:2405.18386*, 2024a.

Yixiao Zhang, Yukara Ikemiya, Gus Xia, Naoki Murata, Marco A Martínez-Ramírez, Wei-Hsiang Liao, Yuki Mitsufuji, and Simon Dixon. Musicmagus: Zero-shot text-to-music editing via diffusion models. *arXiv preprint arXiv:2402.06178*, 2024b.

A QUALITATIVE COMPARISON

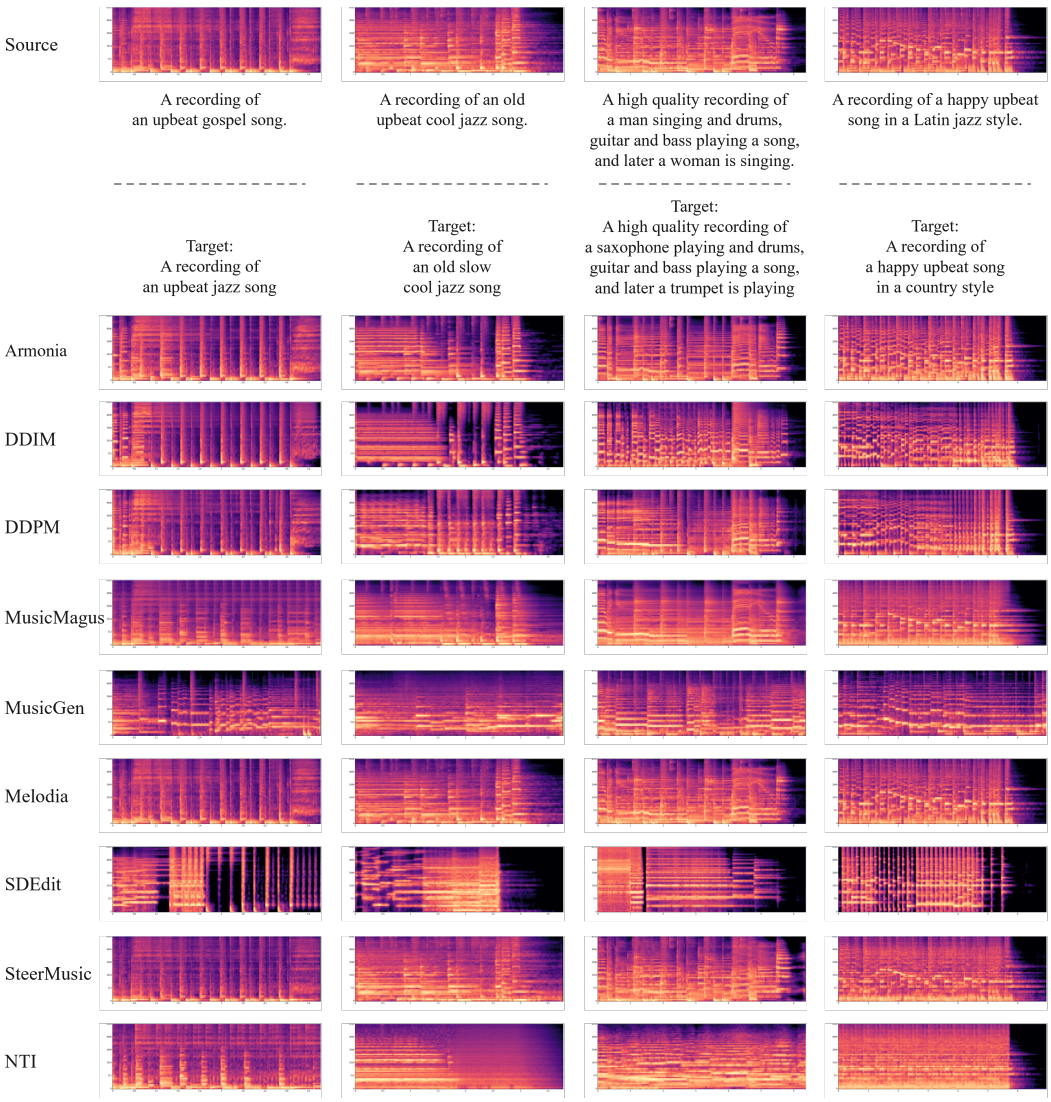

Figure 7: **Qualitative comparison of text-guided music editing.** Four examples cover genre, tempo, and instrument editing. Each column shows the source spectrogram and source/target prompts, followed by outputs from Armonia and eight baselines.

B PSEUDOCODE OF ARMONIA

Algorithm 1 Armonia for Text-Guided Music Editing

Require: Source recording x_{src}; conditions $c_{\mathrm{src}}, c_{\mathrm{tgt}}$; frozen pretrained model
Require: Inversion endpoint T_{start}; selected layers $\mathcal{S}$; temporal weight $w(t)$ and activation threshold τ
Require: Refinement budget K; base learning rate γ; local displacement budget ρ
Ensure: Edited recording x_{edit}

$z_0^{\mathrm{A}} \leftarrow \mathcal{E}(x_{\mathrm{src}})$
for $t = 1, \ldots, T_{\mathrm{start}}$ **do**
  Compute the DDIM proposal b_t from z_{t-1}^{A} under c_{src} using Eq. 4
  $z_t^{\mathrm{A}} \leftarrow b_t$
  if $w(t) \geq \tau$ **then**
    $z \leftarrow b_t$
    for up to K refinement steps **do**
      Compute O_{src}^{ℓ} and O_{mix}^{ℓ} from source and target responses at (z, t), for $\ell \in \mathcal{S}$
      $L \leftarrow \dfrac{1}{|\mathcal{S}|} \displaystyle\sum_{\ell \in \mathcal{S}} \mathrm{MSE}\Big(O_{\mathrm{mix}}^{\ell}, O_{\mathrm{src}}^{\ell}\Big)$
      $u \leftarrow \mathrm{AdamStep}(z, \nabla_z L; \gamma w(t))$
      $z \leftarrow \Pi_{\mathcal{C}_t}(u)$
    end for
    Accept the lowest-loss evaluated candidate as z_t^{A} (Appendix D)
  end if
end for
Obtain z_{edit} by native target-conditioned denoising from $z_{T_{\mathrm{start}}}^{\mathrm{A}}$ under c_{tgt}
$x_{\mathrm{edit}} \leftarrow \mathcal{D}(z_{\mathrm{edit}})$
return x_{edit}

C BASELINE IMPLEMENTATION DETAILS

Unless otherwise stated, each configuration below follows the hyperparameters recommended by the baseline's original paper or released with its official implementation, and is held fixed across MelodiaEdit, MusicDelta, and ZoME-Bench. No baseline hyperparameter was selected or tuned on the evaluation data or against any of the reported metrics, so every method is compared at its recommended operating point. Our main comparison is restricted to zero-shot, training-free methods that require no task-specific training. MusicGen is included as the no-inversion extreme of the design space: it is conditioned on the target text together with a chromagram extracted from the source, and applies no explicit content-preservation control, so it serves as a reference for what text-conditioned generation alone achieves rather than as an editing method with explicit preservation mechanisms.

DDIM Inversion. DDIM Inversion Song et al. (2021) uses 200 steps with full inversion ($T_{\mathrm{start}} = 1000$). The source and target CFG scales are set to 3.0 and 5.0, respectively. Target-conditioned denoising starts from the inverted latent without additional attention control.

DDPM-Friendly. DDPM-Friendly Manor & Michaeli (2024) uses a 200-step diffusion schedule with editing starting at $T_{\mathrm{start}} = 500$. The source and target CFG scales are set to 3.0 and 12.0, respectively. The noise maps recovered through stochastic inversion are reused during editing.

Null-text Inversion. NTI Mokady et al. (2023) is adapted to AudioLDM 2 by optimizing unconditional embeddings against the DDIM reconstruction trajectory. The configuration uses 200 diffusion steps and a CFG scale of 3.5 for reconstruction and editing. Each reconstruction step uses at most 10 Adam updates with a learning rate of 10^{-2}. The optimized embeddings are reused during target-conditioned denoising, with model weights kept fixed.

Melodia. Melodia Yang et al. (2026) uses 200 steps with full DDIM inversion ($T_{\mathrm{start}} = 1000$) and a target CFG scale of 5.5. Self-attention injection is applied to layers 8–14. Source queries and keys replace their target counterparts during denoising, while target values are retained.

MusicMagus. MusicMagus Zhang et al. (2024b) uses 200 steps with full DDIM inversion ($T_{\text{start}} = 1000$). The source and target CFG scales are set to 1.0 and 3.5, respectively. Text-embedding manipulation specifies the attribute change, while cross-attention consistency guides latent updates during denoising.

SDEdit. SDEdit Meng et al. (2021) adds Gaussian noise to the source latent at $T_{\text{start}} = 500$ and performs target-conditioned denoising using a 200-step diffusion schedule. The target CFG scale is set to 12.0.

MusicGen. MusicGen uses the `facebook/musicgen-melody` checkpoint with 1.5B parameters Copet et al. (2023). Generation uses a CFG scale of 3.0 and an output duration of 10 seconds. The model is conditioned on the target text and a chromagram extracted from the source audio.

SteerMusic. SteerMusic Niu et al. (2026) is evaluated using its open-source implementation with AudioLDM 2. The source-initialized latent is optimized for 400 iterations using SGD with an initial learning rate of 0.02. The learning rate is multiplied by 0.9 every 20 iterations. Both source and target CFG scales are set to 30, with a DDS weight multiplier of 2. Diffusion timesteps are sampled uniformly from the normalized interval $[0.02, 0.98]$.

Methods considered but not included. AUDIT Wang et al. (2023) and MelodyEdit Liu et al. (2025) are excluded from the main comparison for two different reasons: AUDIT requires editing-specific training, and MelodyEdit has no public implementation. A fair comparison would require the authors' official code; without it, any number we produce may reflect our re-implementation choices rather than the method itself, so we do not report re-implementations of either method in Table 1, and we make no claim about the performance of either method as reported by its authors. For internal reference only, we did build re-implementations from the papers and run them on the same benchmarks and metrics; these runs are not part of our evidence, and the only conclusion we draw from them is that they did not reveal a ranking reversal large enough to change the conclusions of Table 1. We note that MelodyEdit, which introduced the ZoME-Bench evaluation set used above, is consequently absent from our comparison.

D Armonia Implementation Details

Parameter selection. For MelodiaEdit, the projection budget is selected using the equally weighted mean CLAP score across the five evaluated subsets. The selected value, $\rho = 10^{-6}$, is shared across all five subsets. For MusicDelta, the inversion ratio of 0.8 is selected by mean CLAP from the evaluated ratios between 0.4 and 1.0. ZoME-Bench uses the same configuration as MusicDelta.

Diffusion schedule and state conventions. The inversion endpoint $T_{\text{start}} \leq T$ also specifies the starting timestep of target-conditioned denoising. Algorithm 1 uses the adjacent-timestep notation of the main text. Under timestep subsampling, DDIM updates connect consecutive selected noise levels, and $w(t)$ is evaluated at the actual diffusion timestep. At each inversion step, b_t is computed from the previously accepted state z_{t-1}^{A} and remains fixed as the reference for Local Projection throughout that step. The variable z in the pseudocode abbreviates the current candidate $z_t^{(k)}$.

Attention evaluation and gradient handling. Each evaluation of $\mathcal{L}_{\text{attn}}(z, t)$ recomputes the source- and target-conditioned responses at the same candidate and timestep. The target keys and values are refreshed and detached for the current gradient computation. The source branch remains differentiable with respect to z, including the source queries shared by the source and mixed attention outputs. The mean squared error is averaged over all elements of each layer's output tensor. All pretrained model parameters remain fixed. Gradients are cleared before each refinement iteration, and no gradients are propagated through earlier refinement updates or inversion steps.

Optimizer state and gradient clipping. A fresh Adam optimizer is initialized at each active inversion timestep, with learning rate $\gamma w(t)$. Its moment estimates are retained across refinement iterations within that timestep and reset before refinement at the next active timestep. Before each

Adam update, the latent gradient is clipped using a maximum norm $g_{\max}$. Local Projection is applied after the update, with the current proposal b_t as its fixed center. An out-of-bound candidate is projected radially back toward b_t. Candidates already inside $\mathcal{C}_t$ are retained without modification.

Early stopping. The refinement budget K is an upper bound rather than a fixed number of updates. Let $L_k = \mathcal{L}_{\text{attn}}(z_t^{(k)}, t)$ denote the loss evaluated at the current candidate. For $k \geq 1$, the relative change is computed as

$$r_k = \frac{|L_{k-1} - L_k|}{L_{k-1} + \varepsilon},$$

where ε is a numerical constant. If $r_k < \delta$, refinement stops without taking a further Adam update at that iteration. The threshold δ is a compute-saving device rather than a quality criterion: it stops refinement once the loss has stopped moving, and the accepted candidate is still determined by the selection rule described next.

Candidate selection. Copies of the evaluated candidates are retained whenever a lower attention-distillation loss is observed. After the refinement loop, the final projected candidate is re-evaluated without gradients, with both conditional responses refreshed, so that it also participates in selection. The accepted state z_t^{A} is the candidate with the lowest evaluated loss among those considered at the current timestep, not necessarily the last updated state. This selection uses only $\mathcal{L}_{\text{attn}}$; it does not use listening judgments or downstream editing metrics.

E Objective Metrics Details

CLAP. Audio–text alignment is measured using LAION-CLAP with the HTSAT-base audio encoder and feature fusion disabled Wu et al. (2023). The checkpoint is `music_audioset_epoch_15_esc_90.14.pt`. Edited audio is converted to mono and resampled to 48 kHz. Target text is processed by the RoBERTa tokenizer with a maximum length of 77 tokens. The score is the cosine similarity between the audio and text embeddings. The MelodiaEdit results retain the existing filename-based text format, including the `.wav` suffix.

LPAPS. LPAPS (learned perceptual audio patch similarity), the audio analogue of LPIPS, measures perceptual differences between source and edited audio using the CLAP-based implementation of Manor and Michaeli Manor & Michaeli (2024). Features are extracted from four intermediate blocks of the HTSAT-base encoder using the same music checkpoint as CLAP. Audio is converted to mono and resampled to 48 kHz. Corresponding 10-second windows are compared with 10% overlap. Within each window, features are normalized before computing squared differences. Differences are summed over feature channels, averaged over feature positions, and summed across the four blocks. Window scores are averaged to obtain the audio-pair score.

Chroma. Chroma compares the pitch-class content of source and edited audio using AudioCraft's chroma extractor Copet et al. (2023). For the corrected evaluation, both signals are converted to mono, resampled to 16 kHz, and cropped to their common duration. Features are extracted using a 2,048-point FFT, a 512-sample hop, and twelve pitch classes. Each chroma vector is normalized by its maximum magnitude before computing framewise cosine similarities. Let C^{src} and C^{edit} contain the features extracted from the aligned, unpadded clips, and let N be the smaller of their actual frame counts. The corrected score averages exactly the N frame pairs used in the numerator:

$$\text{Chroma} = \frac{1}{N} \sum_{n=1}^{N} \frac{\langle C_n^{\text{src}}, C_n^{\text{edit}} \rangle}{\max(\|C_n^{\text{src}}\|_2, \varepsilon) \max(\|C_n^{\text{edit}}\|_2, \varepsilon)}, \qquad \varepsilon = 10^{-8}. \tag{12}$$

For batched evaluation, frames introduced solely by batch padding are excluded. The denominator must use the actual number of compared frames after resampling, rather than $\lceil L_{\text{original}}/512 \rceil$ computed from the original sample count. The latter can differ from the number of extracted frames and makes the score depend on the original sample rate.

CQT1-PCC. CQT1-PCC compares the dominant pitch contours of source and edited audio Niu et al. (2026). Audio is resampled to 16 kHz, and the first channel is processed using `nnAudio`'s CQT2010 transform. The transform uses 24 bins per octave, frequency bounds of 65–2100 Hz, and a Hann window. The highest-magnitude frequency bin is selected at each frame. The score is the Pearson correlation between the resulting bin-index sequences, truncated to their common length. Undefined correlations and sequences with fewer than two frames receive a score of zero.

KAD. KAD measures distributional discrepancy using an unbiased Gaussian-kernel maximum mean discrepancy estimator Chung et al. (2025). Features are extracted with the same LAION-CLAP music checkpoint used above. Audio is converted to 48 kHz mono PCM16 and divided into 10-second windows with a 1-second hop. Incomplete trailing windows are zero-padded. The kernel bandwidth follows the median-distance heuristic on edited-audio embeddings. Each reference collection contains unique source recordings, while the comparison collection contains their edited outputs. For MelodiaEdit and ZoME-Bench, KAD is computed within each evaluated subset and averaged equally across the five and three subsets, respectively. For MusicDelta, KAD is computed globally over the evaluated source and edited collections.

F Ablation Study Details

F.1 Configurations of Component Ablation

All component variants use the same evaluation samples across the five MelodiaEdit subsets, with scores averaged equally across subsets. They share a 200-step schedule, full inversion, an empty source prompt, and source and target CFG scales of 3.5. The DDIM reference disables latent refinement and directly performs target-conditioned denoising after inversion. This reference uses the shared ablation settings; the main-table DDIM baseline instead uses source and target CFG scales of 3.0 and 5.0, respectively.

When attention distillation is enabled, refinement uses up-block layers 8–9, Adam with a base learning rate of 10^{-3}, and an inner-loop budget of $K = 3$. The gradient-norm clipping threshold is 0.05, and the relative early-stopping threshold is 10^{-3}. Temporal weighting uses a Gaussian window with $\mu = 800$ and $\sigma = 120$, scales the learning rate by the window weight, and skips refinement when the weight is below 0.1. Disabling temporal weighting applies the constant base learning rate at every inversion step. Local projection bounds the mean-squared latent displacement from the current DDIM proposal by $\rho = 10^{-6}$. Disabling local projection removes this projection while retaining the other optimization settings. Target-conditioned denoising is identical across all variants.

Guidance-objective variants. Table 3 compares the Armonia objective against four alternatives computed from the same forward passes on the same 40 edits. *SA output* is the Armonia objective, i.e. the mean squared error between the mixed and the source self-attention outputs (Eq. 8). *SA map* applies the same criterion to self-attention maps, i.e. to the attention weights, instead of to the attention outputs. *Text-CA output* and *LoA-CA output* apply it to the outputs of the cross-attention layers when these are conditioned on the text prompt and on the language-of-audio (LOA) representation of AudioLDM 2 Liu et al. (2024), respectively. *ENM-style* replaces output matching with the one-step state-matching criterion discussed in Sect. 4.1. The reported values therefore isolate the effect of the matched quantity, holding the remaining refinement settings fixed.

F.2 Temporal Window Configurations

Table 4 summarizes window shape, position, and width at the MelodiaEdit five-subset scale. The underlying sweeps share the seed for each editing pair. Attention distillation and local projection remain enabled, with the shared refinement settings described above.

For the shape comparison, uniform weighting applies the base learning rate of 10^{-3} at exactly the timesteps retained by the default Gaussian window. Gaussian weighting instead scales this learning rate by the window weight. For the position comparison, the center varies over $\mu \in \{600, 800, 1000\}$, with $\sigma = 120$ and an activation threshold of 0.1. For the width comparison, $\sigma \in \{60, 120, 240\}$ with $\mu = 800$ and the same threshold. Windows are clipped to the available inversion timesteps, so changing either the center or width can change both the active-step count and

the effective learning rates. These comparisons therefore evaluate refinement schedules rather than holding the total optimization effort fixed.

At the reported precision, Gaussian weighting improves CLAP, CQT1-PCC, LPAPS, and KAD over uniform weighting within the same active interval, while Chroma is tied. Moving the center to $\mu = 600$ improves preservation but lowers CLAP; $\mu = 1000$ also lowers CLAP, with mixed preservation effects. For width, $\sigma = 60$ has lower CLAP (0.419 versus 0.430) but higher CQT1-PCC (0.241 versus 0.226), lower LPAPS (3.600 versus 3.658), and slightly higher Chroma than the default. Its KAD is higher, so the narrower window does not improve all source-related metrics. The wider $\sigma = 240$ also has lower CLAP (0.427), while CQT1-PCC falls to 0.197 and LPAPS rises to 3.784; Chroma increases slightly, but KAD worsens. The reported values support $(\mu, \sigma) = (800, 120)$ as an alignment-oriented compromise, without establishing a uniformly optimal window.

F.3 Local Projection Budget

Table 5 reports the complete projection-budget comparison on MelodiaEdit. The no-projection row uses the AD+TW reference from Table 2. The budget ρ bounds the mean-squared latent displacement from each DDIM proposal. The sweep uses $K = 3$ and a base learning rate of 10^{-3}. CLAP, LPAPS, and Chroma are averaged equally across the five subsets. All finite budgets have higher CLAP and Chroma and lower LPAPS than the shared no-projection reference. The default $\rho = 10^{-6}$ achieves the highest CLAP and Chroma, while $\rho = 3 \times 10^{-7}$ yields the lowest LPAPS. The best alignment and perceptual source similarity therefore occur at different budgets, rather than improving monotonically with a tighter constraint.

Table 5: Complete projection-budget comparison on MelodiaEdit. Scores are averaged equally across the five subsets. The no-projection reference is shared with Table 2. Bold indicates the best value in each column.

Budget ρ	CLAP ↑	LPAPS ↓	Chroma ↑
10^{-8}	0.425	3.679	0.656
3×10^{-8}	0.425	3.671	0.656
10^{-7}	0.426	3.655	0.659
3×10^{-7}	0.428	**3.644**	0.661
10^{-6} (default)	**0.430**	3.658	**0.662**
3×10^{-6}	0.429	3.726	0.658
10^{-5}	0.425	3.811	0.653
No projection	0.413	3.847	0.649

G Additional Analysis Details

G.1 Hyperparameter Analysis

G.1.1 Inversion Ratio

The inversion ratio specifies the fraction of the diffusion schedule traversed during inversion. The MusicDelta sweep covers ratios from 0.4 to 1.0 in increments of 0.1, using target Armonia with an empty source condition and a 200-step schedule.

Deeper inversion improves target alignment up to a ratio of 0.8, at the cost of increasing LPAPS and decreasing pitch-based similarities. Beyond 0.8, all four metrics deteriorate. The choice of 0.8 therefore prioritizes target alignment, while smaller ratios offer stronger source preservation.

Table 6: Inversion-ratio sensitivity on MusicDelta.

Ratio	CLAP ↑	CQT1-PCC ↑	LPAPS ↓	Chroma ↑
0.4	0.290	**0.619**	**3.589**	**0.332**
0.5	0.315	0.574	3.788	0.328
0.6	0.346	0.508	4.077	0.318
0.7	0.376	0.405	4.459	0.302
0.8 (default)	**0.387**	0.313	4.787	0.287
0.9	0.386	0.288	5.100	0.276
1.0	0.362	0.255	5.287	0.272

Protocol for the remaining refinement sweeps. The K and attention-layer sweeps use the full MelodiaEdit dataset, with scores averaged equally across its five subsets. Learning-rate and CFG sensitivity are analyzed on the Generated Instrument subset, with comparisons made within each table. Configurations within each sweep share the seed for each editing pair. All configurations use full inversion, a 200-step schedule, and an empty source condition. Unless varied, the default uses CFG 3.5, $K = 3$, learning rate 10^{-3}, up-block layers 8–9, and $\rho = 10^{-6}$. The Gaussian window defaults to $(\mu, \sigma) = (800, 120)$ with an activation threshold of 0.1. LPAPS, Chroma, and CQT1-PCC compare each edited output with its source, while KAD compares the edited collection with the unique source clips.

G.1.2 Inner-Loop Budget

Table 7 compares $K \in \{1, 3, 5\}$ on MelodiaEdit. Increasing K from 1 to 3 raises CLAP from 0.419 to 0.430 and lowers LPAPS from 3.961 to 3.658, while also improving CQT1-PCC and Chroma, with a small increase in KAD. The default $K = 3$ achieves the best CLAP, CQT1-PCC, and LPAPS, whereas $K = 1$ gives the lowest KAD. Increasing K further to 5 yields only a small Chroma gain, from 0.662 to 0.664, and worsens the other four metrics, showing no consistent benefit from a larger refinement budget.

Table 7: Inner-loop sensitivity on the full MelodiaEdit dataset. Scores are averaged equally across five subsets; bold indicates the best value in each column.

K	CLAP ↑	CQT1-PCC ↑	LPAPS ↓	Chroma ↑	KAD ↓
1	0.419	0.166	3.961	0.608	**0.050**
3 (default)	**0.430**	**0.226**	**3.658**	0.662	0.052
5	0.429	0.220	3.664	**0.664**	0.057

G.1.3 Attention-Layer Selection

Table 8 compares five attention-layer groups on MelodiaEdit. Up-block layers 8–9 yield the best CLAP, CQT1-PCC, LPAPS, and KAD, while layers 4–5 give slightly higher Chroma. Relative to layers 6–7 the default group improves all five metrics, with LPAPS decreasing from 3.860 to 3.658 and CQT1-PCC increasing from 0.193 to 0.226.

Table 8: Attention-layer selection on the full MelodiaEdit dataset. Scores are averaged equally across five subsets; bold indicates the best value in each column.

Layers	Count	CLAP ↑	CQT1-PCC ↑	LPAPS ↓	Chroma ↑	KAD ↓
Down 1–2	6	0.405	0.148	4.132	0.594	0.081
Mid	3	0.410	0.164	4.083	0.615	0.065
Up 4–5	6	0.408	0.191	3.843	**0.663**	0.079
Up 6–7	6	0.428	0.193	3.860	0.645	0.057
Up 8–9 (default)	6	**0.430**	**0.226**	**3.658**	0.662	**0.052**

G.1.4 LEARNING RATE

The base learning rate is varied over $\{3 \times 10^{-4}, 10^{-3}, 3 \times 10^{-3}\}$. The effective learning rate remains scaled by the Gaussian window. In Table 9, the default 10^{-3} gives the highest CQT1-PCC (0.236) and lowest LPAPS (3.732). Reducing the rate to 3×10^{-4} raises CLAP from 0.458 to 0.460 and lowers KAD from 0.022 to 0.018, at the cost of lower CQT1-PCC and Chroma and higher LPAPS. Increasing it to 3×10^{-3} yields only a small Chroma gain (0.722 to 0.724), while worsening the other four metrics relative to the default. The default therefore favors pitch-contour and perceptual preservation; neither target alignment nor preservation improves monotonically with a larger refinement step.

Table 9: Learning-rate sensitivity on the Generated Instrument subset of MelodiaEdit.

Learning rate	CLAP ↑	CQT1-PCC ↑	LPAPS ↓	Chroma ↑	KAD ↓
3×10^{-4}	0.460	0.196	3.883	0.684	0.018
10^{-3} (default)	0.458	0.236	3.732	0.722	0.022
3×10^{-3}	0.450	0.218	3.796	0.724	0.024

G.1.5 CFG SCALE

The CFG sweep evaluates scales of 2.0, 3.5, and 5.0. Each value is applied to both inversion and target-conditioned denoising. Table 10 shows a consistent alignment–preservation trade-off. Increasing CFG from 2.0 to 3.5 raises CLAP from 0.404 to 0.458, while CQT1-PCC falls from 0.333 to 0.236. The intermediate scale of 3.5 obtains a CLAP gain of 0.054 over 2.0; raising it further to 5.0 lowers CLAP to 0.452, with further losses in all three preservation metrics. KAD decreases from 0.034 to 0.017 across the sweep, despite the deterioration in pairwise source similarity, illustrating that distributional agreement and preservation of individual source clips measure different properties. CFG 3.5 is retained as a compromise between target alignment and source preservation.

Table 10: CFG sensitivity on the Generated Instrument subset of MelodiaEdit.

CFG scale	CLAP ↑	CQT1-PCC ↑	LPAPS ↓	Chroma ↑	KAD ↓
2.0	0.404	0.333	3.188	0.794	0.034
3.5 (default)	0.458	0.236	3.732	0.722	0.022
5.0	0.452	0.174	4.047	0.675	0.017

G.1.6 PROJECTION BUDGET

The complete projection-budget sweep is reported in Appendix F.3. It uses the same five-subset averaging on MelodiaEdit as the K and attention-layer comparisons. The default $\rho = 10^{-6}$ gives the highest CLAP and Chroma; $\rho = 3 \times 10^{-7}$ gives the lowest LPAPS.

G.2 EFFICIENCY AND MEMORY USAGE

All results are evaluated on a single NVIDIA RTX 3090 GPU.

Efficiency of the ablation components. Table 11 compares the five component configurations in Table 2 on the Generated Instrument, Generated Mood, and Generated Style subsets of MelodiaEdit. We average costs within each subset and then equally across the three subsets. Editing time includes inversion, target-conditioned denoising, and waveform decoding, excluding source encoding and the reconstruction branch. Refinement time is the portion of inversion spent in the optimization loop; backward-pass counts reflect early stopping. Peak memory is the mean per-case maximum allocated CUDA memory, excluding unused allocator reserves. These records use the 200-step component-ablation schedule and were collected with shared GPU use.

Temporal weighting reduces the number of active refinement steps and backward passes. With local projection enabled, it reduces active refinement steps from 200 to 91 and average backward passes from 447.68 to 249.01. Recorded editing time falls from 139.41 to 81.94 seconds, a 41.2% reduction,

while refinement time falls from 111.98 to 54.63 seconds. Without projection, temporal weighting also lowers editing time, from 151.94 to 84.33 seconds. Adding projection to the windowed configuration changes recorded time only modestly, from 84.33 to 81.94 seconds, alongside a decrease in backward passes from 259.93 to 249.01. Peak memory remains approximately 5.4 GiB across the attention-distillation variants, compared with 3.95 GiB without refinement: temporal weighting mainly reduces repeated computation, with little effect on peak memory.

Table 11: Component efficiency on the three generated MelodiaEdit subsets. AD, TW, and LP follow Table 2.

Components			Mean cost per edit				
AD	TW	LP	Editing (s)	Refinement (s)	Active steps	Backward passes	Peak (GiB)
×	×	×	28.99	0.00	0	0.00	3.95
✓	×	×	151.94	123.57	200	568.49	5.41
✓	✓	×	84.33	56.33	91	259.93	5.40
✓	×	✓	139.41	111.98	200	447.68	5.43
✓	✓	✓	81.94	54.63	91	249.01	5.43

End-to-end comparison. Table 12 uses seven common 10-second MusicDelta inputs, full inversion with a 100-step schedule, FP16, and batch size one on dedicated RTX 3090 GPUs. Each method receives two excluded warmup runs. Synchronized wall-clock timing covers input loading, preprocessing, conditioning, inversion, editing, decoding, and CPU waveform return; model loading and output-file writing are excluded. Peak memory is the mean per-edit maximum allocated CUDA memory in GiB.

Armonia takes 44.52 seconds per edit, compared with 106.20 seconds for NTI and 19.16 seconds for the DDIM reference. It therefore uses 58.1% less time than NTI but takes 2.32 times as long as DDIM in this benchmark.

Table 12: Runtime and peak GPU memory for 10-second audio. Diffusion methods use a 100-step schedule on RTX 3090 GPUs.

Method	Inversion/Forward (s)	Editing/Decoding (s)	Total (s)	Peak memory
DDIM Inversion	9.81	9.35	19.16	4.50
DDPM-Friendly	10.01	9.59	19.60	4.52
Melodia	10.13	9.88	20.01	9.62
MusicMagus	44.33	36.09	80.42	8.69
SDEdit	0.50	9.50	10.00	4.49
MusicGen	–	17.93	17.93	4.21
SteerMusic	4.26	51.97	56.23	4.64
NTI	100.57	5.63	106.20	3.67
Armonia	36.83	7.69	44.52	5.51

H Subjective Evaluation Details

H.1 Metric Definitions

Participants independently rate relevance (REL), consistency (CON), and overall musical quality (OMQ) on a five-point scale, with higher scores indicating better performance.

Relevance (REL). REL assesses whether the edited audio achieves the change specified by the target prompt. Participants answer: *"To what extent does the edited audio achieve the requested change?"* A high score indicates that the requested change is clearly audible and consistently realized.

Consistency (CON). CON assesses the preservation of the source's pitch contour and musical structure. Participants answer: *"Apart from the attributes that the instruction asks to change, to what*

extent does the edited audio preserve the musical content and performance structure of the source?" For instrument changes, the assessment focuses on melody, rhythm, and phrasing. Changes explicitly requested by the instruction are not penalized.

Overall Musical Quality (OMQ). OMQ assesses the clarity, naturalness, and coherence of the edited audio. Participants answer: *"How clear, natural, and coherent does the edited audio sound, considering distortion, noise, and unnatural interruptions?"* A high score indicates a coherent musical result with few audible defects. This dimension concerns audible quality rather than personal preference for the musical style.

H.2 Experimental Procedure

Participants and samples. The study involves 67 participants and 30 editing cases, with ten cases from each dataset. Each participant evaluates four anonymized method outputs for each assigned case. Method assignments are balanced across participants to distribute the listening workload.

Audio presentation. The source audio and editing instruction are presented alongside the edited outputs. For each case, all methods use the same temporal excerpt. Case order and output positions are randomized. Participants can replay the source and outputs when making comparisons.

Rating procedure. Participants provide a separate score for each of the three dimensions. They assess the requested change for REL, compare the retained musical content with the source for CON, and judge the edited audio's overall quality for OMQ. An "unable to judge" response is recorded as missing rather than assigned a midpoint score.

H.3 Evaluation Interface

The instruction page introduces the source audio, source and target prompts, and edited output, and asks participants to use headphones (Figure 8). The listening interface presents both prompts alongside audio players for the source and edited result (Figure 9). The questionnaire provides a separate question and brief guidance for REL, CON, and OMQ, each rated from 1 (Bad) to 5 (Excellent) (Figure 10).

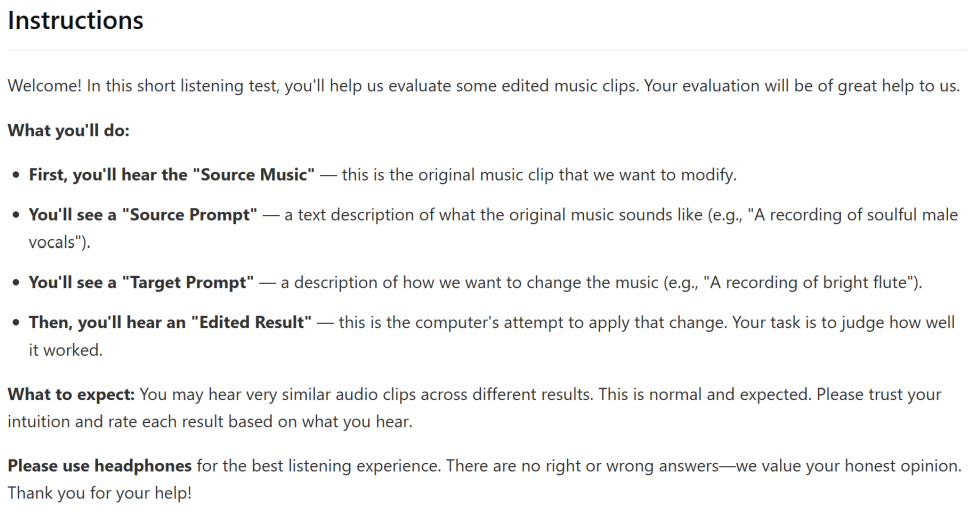

Figure 8: Instruction page introducing the listening task and recommending headphone use.

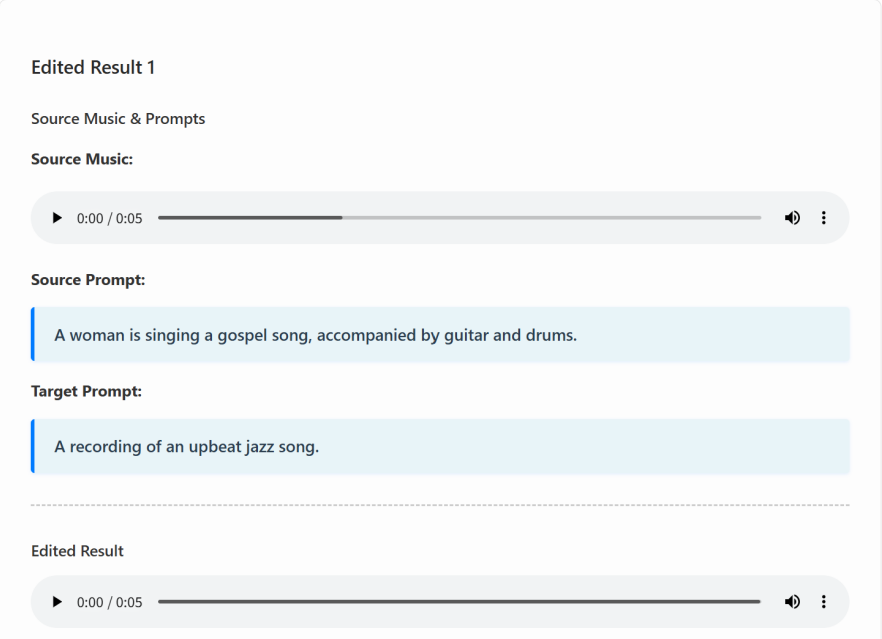

Figure 9: Listening interface displaying the source and target prompts with audio players for the source and edited result.

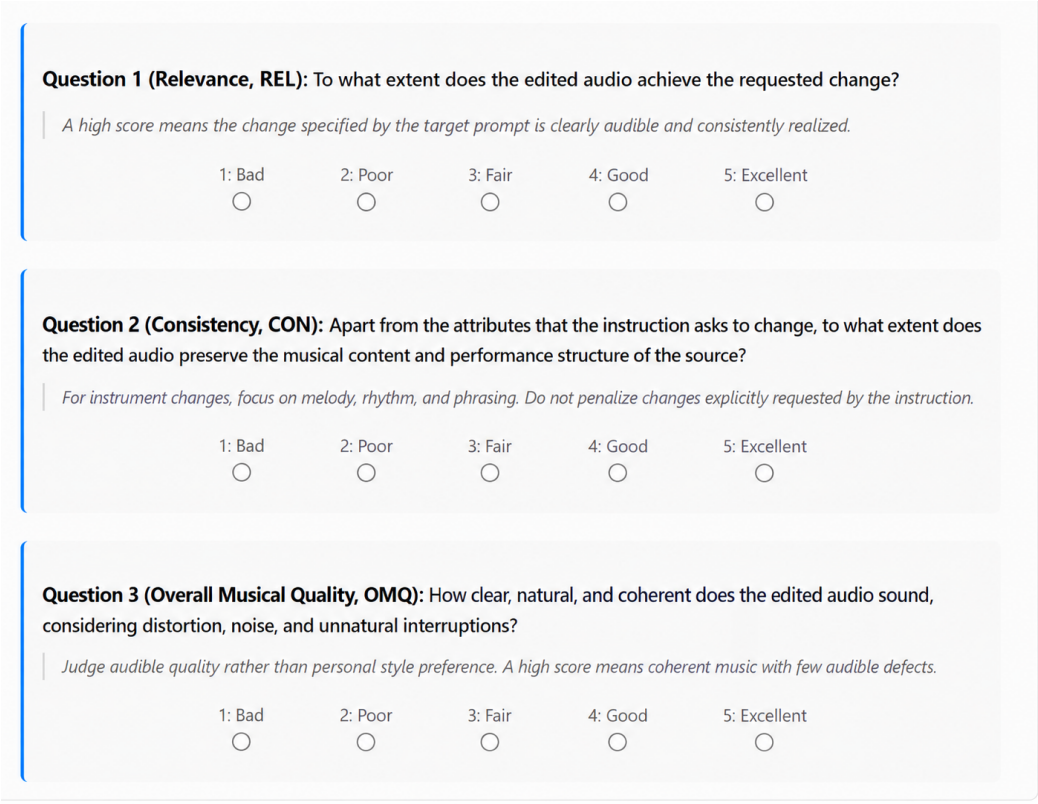

Figure 10: Evaluation questionnaire with separate five-point ratings for relevance (REL), consistency (CON), and overall musical quality (OMQ).