[Supplementary Materials] SteerMusic: Enhanced Musical Consistency for Zero-shot Text-Guided and Personalized Music Editing

Contents

1 MORE BACKGROUND

1.1 Score Distillation Sampling

In Score Distillation Sampling (SDS), a pretrained, frozen diffusion model is employed to estimate the score—i.e., the gradient of the log-density—of the conditional distribution $p(x \mid y)$. The key idea is to optimize a generator function

$$x = g(\theta)$$

with respect to θ so that the generated data (e.g., an image or an audio) x attains high likelihood under the diffusion model's learned density. To this end, we define a differentiable loss $\mathcal{L}_{\text{SDS}}$ whose minimization produces samples resembling those from the diffusion model.

$$\mathcal{L}_{\text{Diff}}(\phi, x = g(\theta)) = w(t)\|\epsilon_\phi(x_t, y, t) - \epsilon\|_2^2$$

In effect, we solve

$$\theta^* = \arg\min_\theta L_{\text{Diff}}(\phi, x = g(\theta)),$$

where $\mathcal{L}_{\text{Diff}}(\phi, x)$ is the original diffusion training loss used to learn $p(x \mid y)$, and ϕ denotes the parameters of the frozen diffusion model.

More precisely, the gradient of the diffusion loss with respect to θ is given by

$$\nabla_\theta \mathcal{L}_{\text{Diff}}(\phi, x = g(\theta)) = \mathbb{E}_{\epsilon,t}\left[w(t)\left(\epsilon_\phi(x_t, y, t) - \epsilon\right) \cdot \underbrace{\frac{\partial \epsilon_\phi(x_t, y, t)}{\partial x_t}}_{\text{Jacobian}} \cdot \frac{\partial x_t}{\partial \theta} \right].$$

Since computing the U-Net Jacobian $\frac{\partial \epsilon_\phi}{\partial x_t}$ is computationally expensive and poorly conditioned at low noise levels, we omit this term [14]. The simplified gradient becomes

$$\nabla_\theta \mathcal{L}_{\text{SDS}}(\phi, x = g(\theta)) \approx \mathbb{E}_{\epsilon,t}\left[w(t)\left(\epsilon_\phi(x_t, y, t) - \epsilon\right) \cdot \frac{\partial x_t}{\partial \theta} \right].$$

Intuitively, this update nudges x in a direction that increases its (conditional) likelihood according to the diffusion model's learned score function.

1.2 Delta Denoising Score

In image domain, using SDS to perform image editing directly suffers blurry issues [6], where the gradient of vanilla SDS can be decomposited into two components:

$$\nabla_\theta \mathcal{L}_{SDS}(x, y, \epsilon, t) := \delta_{\text{text}} + \delta_{\text{bias}} \tag{1}$$

where component δ_{text} is a desired direction that directs the optimization to match the condition y (i.e., y is a target prompt in the editing setting), and δ_{bias} is undesired component which causes unintended editing on the results such as blurry and smooth.

In the image editing task, given matched and unmatched image-prompt data pairs $\{x^{\text{src}}, y^{\text{src}}\}$ and $\{x, y^{\text{tgt}}\}$, respectively. The delta denoising loss can be formulated as

$$\begin{aligned}\mathcal{L}_{\text{DD}}(\phi, x, x^{\text{src}}, y^{\text{src}}, y^{\text{tgt}}) \\ = \mathbb{E}_{\epsilon,t}[w(t)\|\epsilon_\phi(x_t, y^{\text{tgt}}, t) - \epsilon_\phi(x_t^{\text{src}}, y^{\text{src}}, t)\|_2^2]\end{aligned} \tag{2}$$

where x_t and x_t^{src} shares the same sampled noise ϵ.

Same as in SDS, by omitting the Jacobian over the diffusion model, the gradient over the geneator parameter θ is given by

$$\nabla_\theta \mathcal{L}_{\text{DDS}} = \mathbb{E}_{\epsilon,t}[w(t)(\epsilon_\phi(x_t, y^{\text{tgt}}, t) - \epsilon_\phi(x_t^{\text{src}}, y^{\text{src}}, t))\frac{\partial x}{\partial \theta}] \tag{3}$$

By adding and subtracting ϵ in Eq. 3, the DDS can be represented as a difference between two SDS scores:

$$\nabla_\theta \mathcal{L}_{\text{DDS}} = \nabla_\theta \mathcal{L}_{\text{SDS}}(x, y^{\text{tgt}}) - \nabla_\theta \mathcal{L}_{\text{SDS}}(x^{\text{src}}, y^{\text{src}}) \tag{4}$$

Thus, [6] claimed the non-zero gradient of the second term in Eq. 4 can be attributed to the noisy direction

$$\nabla_\theta \mathcal{L}_{\text{SDS}}(x^{\text{src}}, y^{\text{src}}) \approx \delta_{\text{bias}} \tag{5}$$

By subtracting the bias term, DDS can be considered a distilled direction that concentrates on editing the relevant portion of the inputs (i.e., image) to match to the target prompt y^{tgt}.

1.3 Denoising Diffusion Implicit Model

Given a diffusion probabilistic model parameterized by ϕ and a diffusion process defined as $q(x_t|x_0) := \mathcal{N}(x_t; \sqrt{\alpha_t}x_0, (1-\alpha_t)I)$, where the α_t represents the variance of the forward diffusion process at time step t, x_t represents the noised latent representation of the data x_0. The DDIM [15] defines a update rule in the reverse diffusion process, which the formulation is given by

$$x_{t-1} = \underbrace{\sqrt{\alpha_{t-1}}\left(\frac{x_t - \sqrt{1-\alpha_t}\epsilon_\phi^{(t)}(x_t)}{\sqrt{\alpha_t}}\right)}_{\text{'predicted } x_0\text{'}} + \underbrace{\sqrt{1-\alpha_{t-1}-\sigma_t^2}\epsilon_\phi^{(t)}(x_t)}_{\text{'direction pointing to } x_t\text{'}} + \underbrace{\sigma_t\epsilon_t}_{\text{'random noise'}} \tag{6}$$

where σ_t is a free variable that controls the stochasticity in the reverse process.

DDIM Inversion. By setting σ_t to 0, we can obtain a deterministic update rule which can be reversed to a deterministic mapping between x_0 and its latent representation x_T. The inverse mapping is refered as DDIM inversion, which is formulated as

$$\frac{x_{t+1}}{\sqrt{\alpha_{t+1}}} - \frac{x_t}{\sqrt{\alpha_t}} = \left(\sqrt{\frac{1-\alpha_{t+1}}{\alpha_{t+1}}} - \sqrt{\frac{1-\alpha_t}{\alpha_t}}\right)\epsilon_\phi^{(t)}(x_t) \tag{7}$$

1.4 Classifier-free Guidance

Given a diffusion model jointly trained on conditional and unconditional embeddings. In the sampling phase, samples can be generated using classifier-free guidance (CFG) [7]. The prediction with the conditional and unconditional estimates are defined as following equation

$$\epsilon_\phi^\omega(x_t, h_t) := \omega\epsilon_\phi(x_t, y, t) + (1-\omega)\epsilon_\phi(x_t, \varnothing, t) \tag{8}$$

where ω is the guidance scale that controls the trade-off between mode converage and sample fidelity, and $\varnothing$ is a null token used for unconditional prediction.

2 USER STUDY

2.1 User Preference Study for Model Comparison

We follow the design of [10] for this user preference study. This user preference study contains two parts, the first part is for evaluating text-guided music editing methods and the second part is for evaluating the personalized music editing methods. We randomly select 10 source musics with corresponding source and target prompts from ZoME-Bench [9] dataset in this user study, each music has 10 seconds duration. For each question, we provide two edited music, one is obtained by our method and the other one is obtained by the compared method, users are asked to select the best matched edited music according to the question. We distribute this user study questionnaire to some open-public groups who are interested in music and have at least one year music training. The order of questions and edited samples are also randomly shuffled in our questionnaire.

For the first part, we include 20 questions with 10 source musics and compare with two methods (i.e., MusicMagus [20] and ZETA [10]). For each question, we provide a source music, a edit instruction and two edited results. We ask users to select the best-matched result from the two provided results according to the question. Figure 1 and Figure 2 demonstrate the tutorial to guide users to select the best matched choice before the main listening test of text-guided music editing, and an exact sample question in this user study.

For the second part, we include 20 questions with 10 source musics and compare with two methods (i.e., DreamSound [13] and Textual inversion [13]). For each question, we provide a source music, a edit instruction, a reference music for the target style, and two edited results. We ask users to select the best-matched result from the two results provided according to the question. Figure 3 and Figure 4 demonstrate the tutorial to guide users to select the best matched choice before the main listening test of personalized music editing, and an exact sample question in this user study.

This user study is anonymous, before the user study, participants were asked to provide their age and number of years for music training.

2.2 Mean Opinion Score Study

In order to test the objective metric sensitivity, we conduct additional mean opinion score (MOS) study to further verify our method compared to the baselines for source music correspondence and target style consistency subjectively. Similar as the user preference study in Section 2.1, the MOS study contains two parts: The first part is to verify SteerMusic with 4 baselines (DDIM [15], SDEdit [11], ZETA [10], and MusicMagus [20]), which contains 5 randomly selected source music with edited results. The second part is to verify SteerMusic+ with 2 baselines (Textual inv. and DreamSound [13]), which contains 5 randomly selected source music with edited results. Each music sample has 10-second duration, the MOS study test takes approximate 15 minutes to be completed. The order of questions and edited samples are randomly shuffled in our questionnaire. We distribute this user study questionnaire to some open-public groups who are interested in music and have at least one year of music training.

Each of the edited results is followed by two questions:

(1) Please rate how well the content (e.g., melody and vocal elements) remains consistent with the source music.
(2) Please rate how well the edited result matches the target style.

Participants were asked to give their rate from 1- Bad to 5-Excellent. Example questions for part 1 and part 2 can be found in Figure 5 and Figure 6. We collected 23 complete responses for Part 1 and 20 full responses for Part 2 from participants with at least 1 year and on average 3 years of music training experience.

3 EXPERIMENTAL DETAILS

3.1 Evaluation Metrics

In our experiment for zero-shot text-guided music editingt task, we follow [4, 5, 10] , and use the "music_audioset_epch_15_esc_90.14.pt"

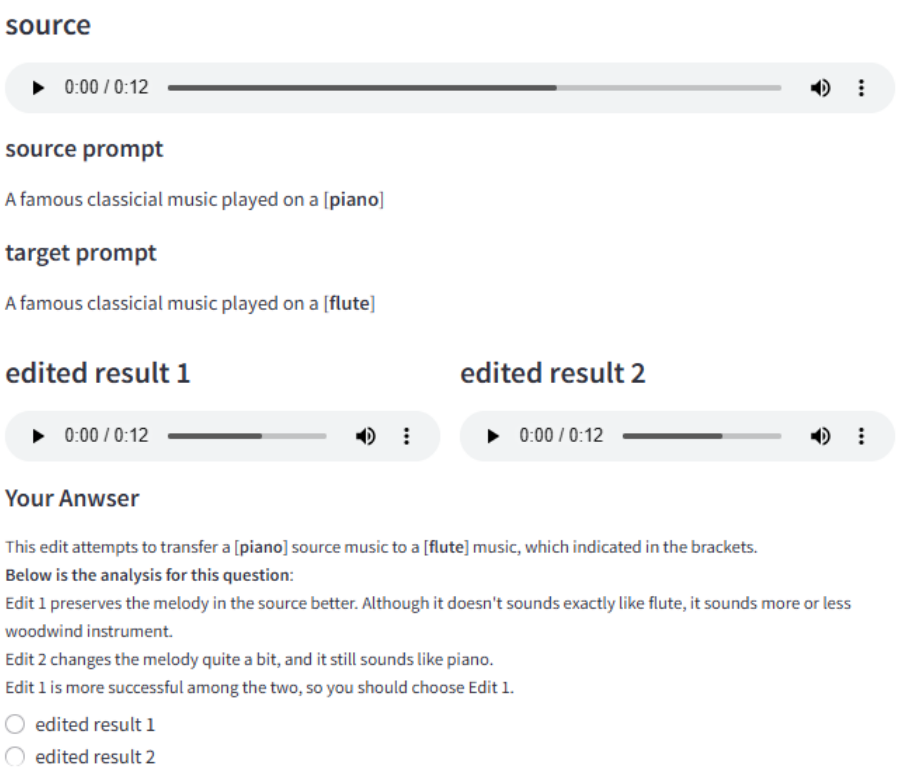

Figure 1: The tutorial for a sample question before the text-guided music editing listening test.

checkpoint of LAION-AI [2, 17] to calculate the CLAP score between target prompts and edited music. Since ZoME-banch [9] dataset contains music clips with 10-second duration, and since this checkpoint was trained for 10-second long segments. We do not apply windows when calculating the CLAP score.

We use CQT2010 function in nnAudio library [1] to calculate CQT features, where we set n_bins = 128 an bins_per_octave=24 under 16000 Hz sampling rate. For the CQT-1 PCC metric, we follow [8] and extract the top 1 CQT bins where contains the most of melody information. The detail CQT-1 PCC metric can be formulated as

$$\text{CQT-1 PCC} = \frac{\sum_i^T (c_i^{src} - \bar{c}^{src})(c_i^{tgt} - \bar{c}^{tgt})}{\sqrt{\sum_i^T (c_i^{src} - \bar{c}^{src})^2 \sum_i^T (c_i^{tgt} - \bar{c}^{tgt})^2}} \tag{9}$$

where c_i is the ith index of CQT-1 value.

3.2 Detail Experimental Results of SteerMusic+ Cross Music Concepts

Table 1 and Table 2 provide detailed results of the model comparison for different concepts of musical instruments and music genre. According to the tables, SteerMusic+ outerperforms the baseline methods cross different musical concepts, indicating its superiors for a higher edit fidelity on personalized music editing with enhanced instruction-irrelevent source music content consistency.

[1] https://github.com/KinWaiCheuk/nnAudio

In Table 1, we include an extra objective metric that calculates cut-off MFCCs cosine similarity (MFCCs COS) between edited music and reference music. Following [3], we design this metric as additional objective metric to evaluate perceptual timbre similarity between edited results and reference music, where the metric is given by

$$\text{MFCCs COS} = \cos(f_{c:13}^{\text{tgt}}, f_{c:13}^{\text{ref}}) \tag{10}$$

where f is a cut-off MFCCs feature of musical signal x, c represents the cut-off frequency bins. We set $c = 3$ in our experiment. By excluding the lower frequency bins of the MFCCs, which primarily capture pitch and note-related information, the higher frequency bins can be emphasized to better capture timbre characteristics. The MFCCs COS metric can potentially measures the timbre similarity.

3.3 More visualization for SteerMusic+

Figure 9 presents an additional visual comparison between SteerMusic+ and other baseline methods (DreamSound and Textual inversion) across various musical style concepts on the same source music, further highlighting the superiority of SteerMusic+ in preserving music content while achieving high edit fidelity aligned with the target concept.

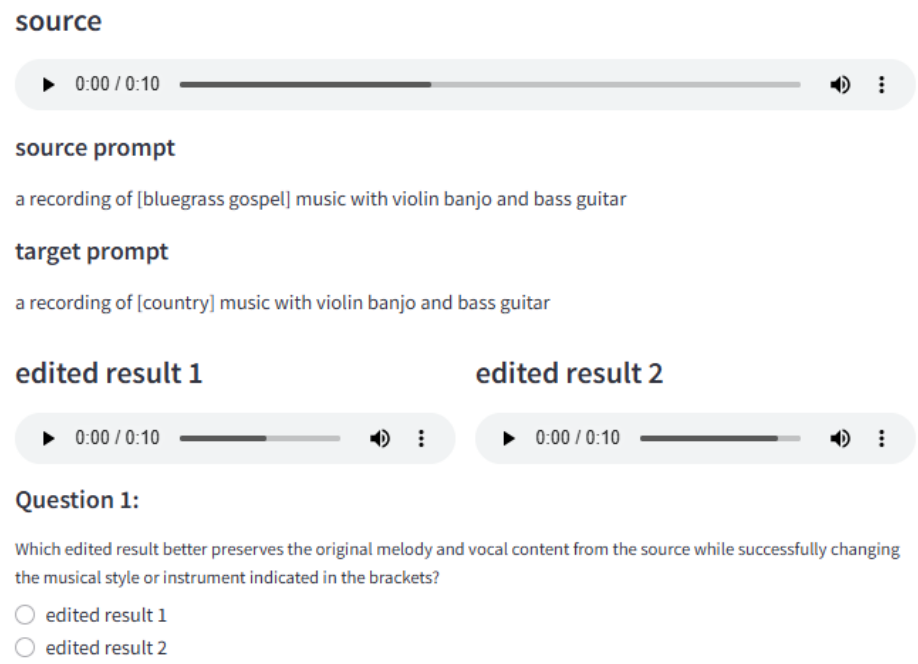

Figure 2: A sample question for the text-guided music editing listening test.

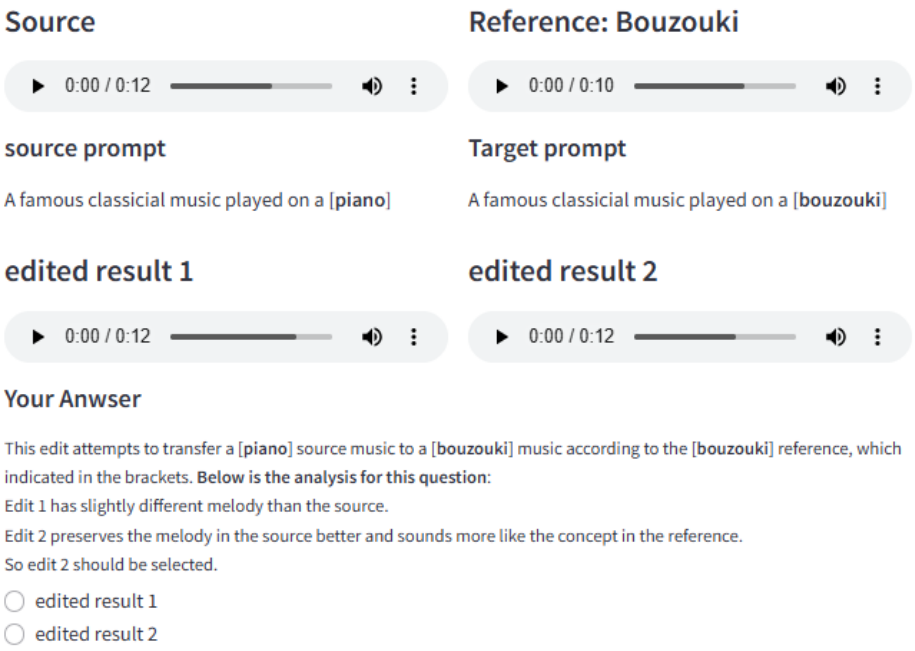

Figure 3: The tutorial for a sample question before the personalized music editing listening test.

4 CLASSIFIER-FREE GUIDANCE STRENGTH

4.1 CFG Strength for SteerMusic

Following [6], where a higher CFG value leads to faster optimization convergence, we conducted an ablation study on SteerMusic using varying CFG values and DDS gradient scales, as shown in Figure 7. All experiments were run for 400 optimization steps.

We observe that lower CFG values (e.g., 5) result in lower CLAP scores, especially when using the same variance scale $w(t)$. This

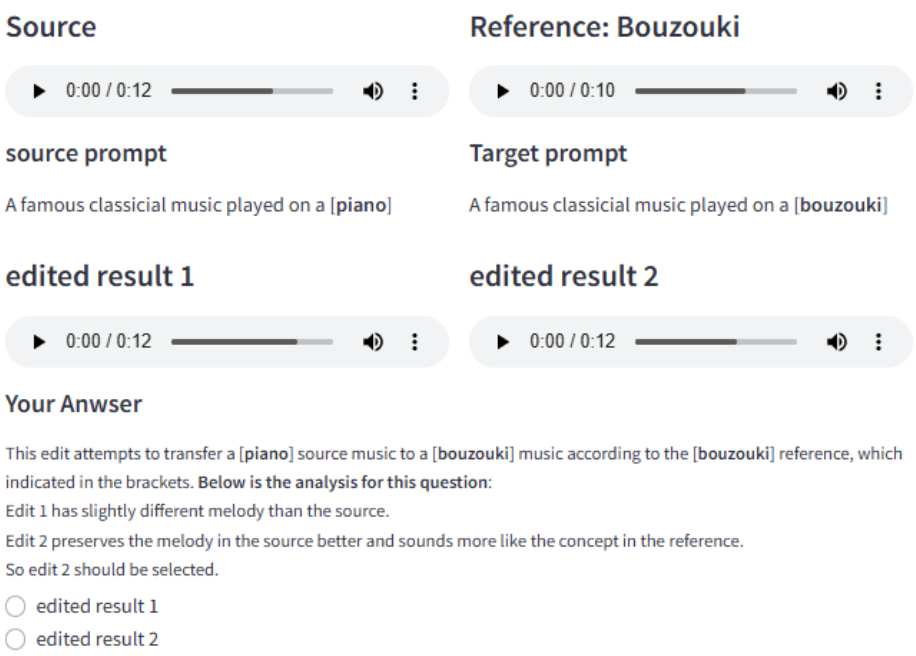

Figure 4: A sample question for the personalized music editing listening test.

suggests that the edited outputs remain closer to the source music, achieving higher consistency but at the cost of weaker alignment with the target prompt. As the CFG increases, the model places more emphasis on the target prompt, resulting in higher CLAP scores but also an increase in LPAPS, indicating a degradation in structural consistency with the source. This trade-off becomes more pronounced with larger DDS gradient scales (e.g., $5 \times w(t)$), where the edited results aggressively deviate from the source, leading to a steep rise in LPAPS despite better CLAP alignment. We find that moderate CFG values (e.g., 15–30) offer a better balance between style adaptation and source preservation, especially under lower DDS scaling. However, beyond a certain threshold (e.g., CFG = 50), especially at high $w(t)$, the results become over-edited, causing a sharp increase in LPAPS and instability in content preservation. Interestingly, we also find increase the gradient scale during optimization a bit (e.g., $2 \times w(t)$) helps to further enhance the optimization convergence. These results highlight the importance of carefully tuning both CFG and DDS weight scaling to balance semantic alignment and source music content preservation during text-guided music editing.

4.2 CFG Strength for SteerMusic+

In this study, we conduct an ablation study for CFG strength for SteerMusic+ on the personalized music editing task. As shown in Figure 8, we study how CFG value affects the performance on SteerMusic+. All experiments were run for 400 optimization steps on a personalized diffusion model fine-tuned on [bouzouki] musical concept.

According to Figure 8 (a) and (b), under the same optimization steps, the CFG values controls the closeness of edited results to the target concept as the higher CFG values leading to a lower CDPAM score. However, as we mentioned the experiment section in our main text, it is a trade-off between style consistency and source music content preservation (indicated by CQT-1 PCC values in Figure 8 (a) and LPAPS score in Figure 8 (b)). In our experiment, we set GFG = 15 on SteerMusic+ for the task of personalized music editing. These results highlight the importance of carefully tuning CFG weight scaling to balance semantic alignment and source music content preservation during personalized music editing.

5 MORE EXPERIMENT AND DISCUSSION FOR STEERMUSIC ADAPTATION

In this section, we further explore the adaptation of variant score distillation methods within the SteerMusic framework for zero-shot text-guided music editing task. Specifically, we investigate two approaches: the first involves directly adapting the score distillation sampling (SDS) method [14], as formulated in Eq. 1.1, for zero-shot text-guided music editing. The second approach leverages an improved variant of the DDS method, originally proposed for text-guided image editing, known as Contrastive Denoising Score (CDS)[12].

5.1 Score Distillation Sampling for Zero-shot Text-guided Music Editing

In our first attempt, we directly adapt vanilla score distillation sampling (SDS) [14] method for text-guided music editing, which the gradient over θ is given by

$$\nabla_\theta \mathcal{L}_{\text{SDS}}(x, y^{\text{tgt}}, \epsilon, t) = \mathbb{E}_{\epsilon,t}[w(t)(\epsilon_\phi(x_t, y^{\text{tgt}}, t) - \epsilon)\frac{\partial x}{\partial \theta}] \quad (11)$$

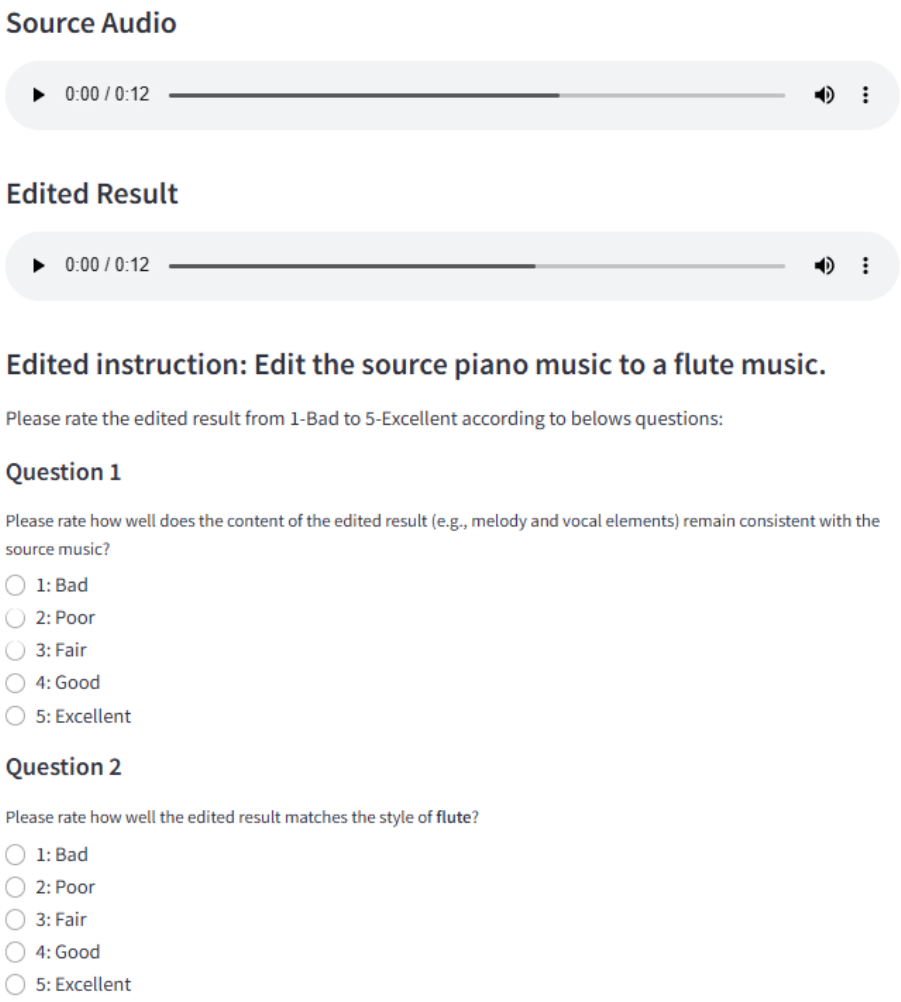

Figure 5: A sample question for MOS study test (Part1) for SteerMusic.

where $\epsilon \sim \mathcal{N}(0, I), t \sim \mathcal{U}(1, T)$.

5.2 SteerMusic with Contrastive Loss Regularization

In our second attempt, we draw inspiration from Contrastive Denoising Score (CDS) [12] by incorporating an additional contrastive loss regularization to further enhance source music consistency. The CDS method was originally proposed to solve the limitation of DDS that cannot maintain spatial structure consistency in edited images. We coin the variate SteerMusic method with additional contrastive loss regularization as SteerMusic$^{\diamond}$.

Inspired by [12], the desired edited results should not only align well with the target prompt, but also incorporating other music structural elements such as melody and harmony of the input source music. Motivated by [9] that uses self-attention queries to refine musical structures during editing. Recent studies in image domain shows that self-attention features of text-to-image diffusion models are embedded with detailed spatial information, which allows to build image semantic correspondence using these features [1, 16, 18, 19]. Self-attention features in audio generative diffusion models also indicates an overall audio structures [9]. To this end, we adopt CDS method [12] and we include a patchwise contrastive loss between on self-attention features into SteerMusic, which further enhances the source music structures on edited results.

During DDS gradient computing process, we extract self-attention features as $\hat{h}_l$ and h_l, where h_l and $\hat{h}_l$ represents the intermediate features passed through the residual block and self-attention block conditioned on y^{tgt} and y^{src}, respectively. Unlike PCon loss in SteerMusic+, we keep the original size of self-attention features which have shape as $\mathbb{R}^{(T_l \times F_l) \times C_l}$, where T_l, F_l, and C_l represents the size of temporal, spatial and channel dimension in the l-th layer, respectively. The query patch is sampled from the feature map h_l. We denote $s \in \{1, 2, ..., S_l\}$ is the query patch, where $S_l = T_l \times F_l$. For each query, the patch at the corresponding spatial location on the feature map $\hat{h}_l$ is 'positive' and the non-corresponding patches within the feature map as 'negative'. The positive patch is referred

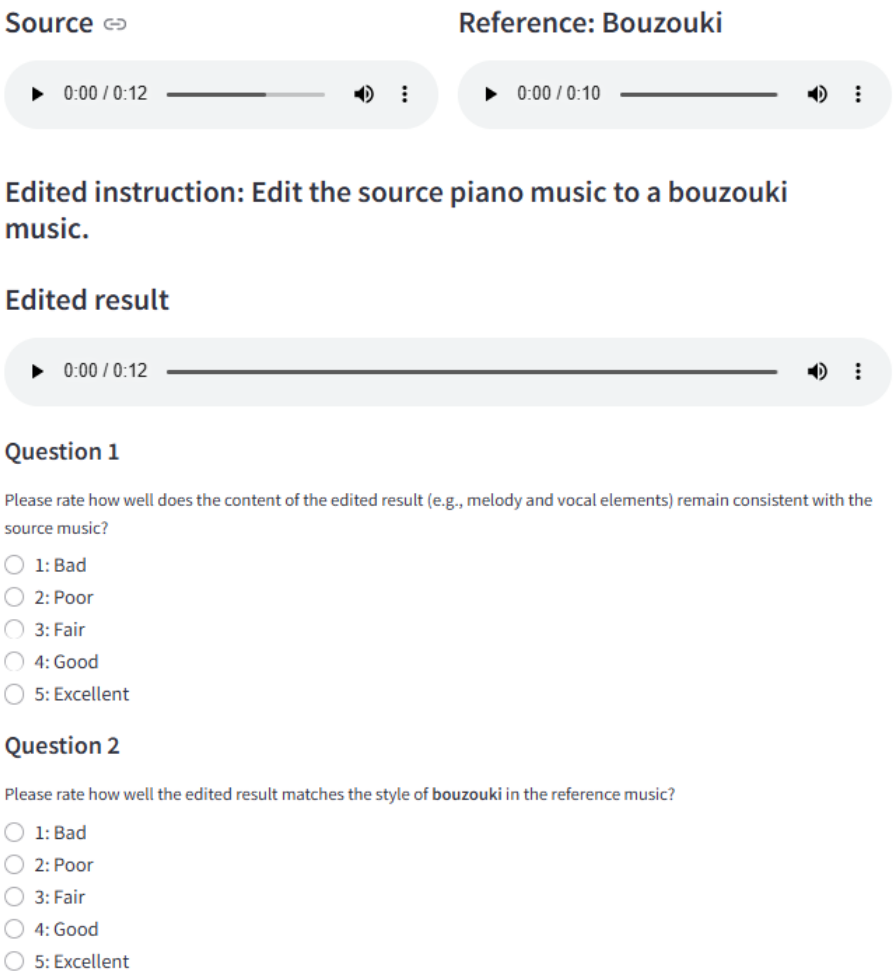

Figure 6: A sample question for MOS study test (Part2) for SteerMusic+.

Table 1: Model comparison on personalized music instrument transfer (SteerMusic+ uses the same personalized model as DreamSound).

Method	Concept	FAD_{CLAP} ↓	$FAD_{Viggish}$ ↓	CQT-1 PCC ↑	LPAPS ↓	MFCCs COS ↑	CDPAM ↓
Textual inv.	Guitar	0.565	2.464	0.148	5.341	**0.666**	0.813
DreamSound	Guitar	0.683	3.454	0.247	4.949	0.647	0.739
SteerMusic+	Guitar	**0.358**	**0.398**	**0.425**	**3.963**	0.637	**0.711**
Textual inv.	Ocarina	0.490	1.706	0.184	5.261	**0.094**	0.998
DreamSound	Ocarina	0.714	2.063	0.347	4.976	-0.097	0.922
SteerMusic+	Ocarina	**0.341**	**0.385**	**0.493**	**3.913**	0.045	**0.919**
Textual inv.	Bouzouki	0.450	1.739	0.193	5.274	0.576	0.452
DreamSound	Bouzouki	0.577	1.309	0.385	4.750	0.761	0.441
SteerMusic+	Bouzouki	**0.358**	**0.651**	**0.439**	**4.165**	**0.773**	**0.440**
Textual inv.	Sitar	0.526	1.660	0.206	5.218	0.297	0.376
DreamSound	Sitar	0.770	2.969	0.230	5.303	0.772	0.279
SteerMusic+	Sitar	**0.450**	**0.755**	**0.266**	**4.509**	**0.830**	**0.229**

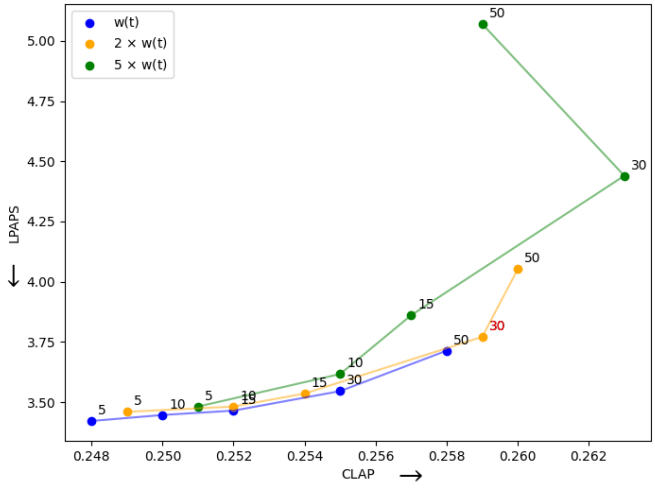

Figure 7: Ablation study of SteerMusic analyzing the trade-off between style correspondence (CLAP) and source music content consistency (LPAPS) under varying classifier-free guidance (CFG) values under 400 optimization steps. Results are shown for three levels of weight scaling on the weighting function $w(t)$: 1×, 2×, and 5×. Increasing CFG improves alignment with the target prompt (higher CLAP) but often at the cost of higher LPAPS, indicating reduced structural fidelity to the source. In our experiment, we use CFG=30 with 2 times $w(t)$.

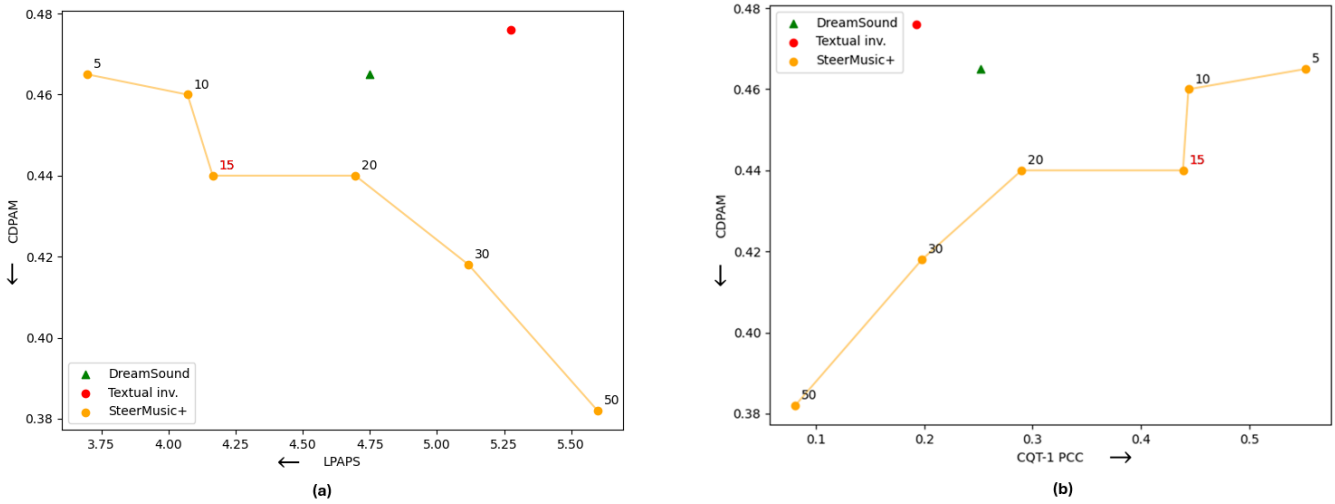

Figure 8: Ablation study of SteerMusic+ analyzing the trade-off between style correspondence and source music melody consistency under varying classifier-free guidance (CFG) values under 400 optimization steps. Increasing CFG values push the edited result closer to the target concept with lower CDPAM; however, it also causes loss source music content (e.g., melody) with higher LPAPS and lower CQT-1 PCC socre.

as $\hat{h}_l^s$ and the other patches as $\hat{h}_l^{S_l \setminus s}$). The additional PatchNCE loss function is formally defined as

$$\mathcal{L}_{\text{PatchNCE}}(x, x^{src}) = \mathbb{E}_h[\sum_l \sum_s \ell(h_l^s, \hat{h}_l^s, \hat{h}_l^{S_l \setminus s})] \tag{12}$$

$$\ell(h, h^+, h^-) = -\log(\frac{\exp(h \cdot h^+/\tau)}{\exp(h \cdot h^+/\tau) + \exp(h \cdot h^-/\tau)}) \tag{13}$$

where $\exp(h \cdot h^+/\tau)$ is positive sample that with the same patch location, $\exp(h \cdot h^-/\tau)$ is negative sample with mismatched spatial location in the self-attention features, τ is a temperature parameter as $\tau > 0$. Following [12], the gradient of $\mathcal{L}_{\text{PatchNCE}}(x, x^{src})$ loss will propagate to the hidden state of self-attention layers h to regularize $\mathcal{L}_{\text{DDS}}$ to have overall content consistency between x and x^{src}.

The function of $\mathcal{L}_{\text{PatchNCE}}(x, x^{src})$ in SteerMusic is fundamentally different to $\mathcal{L}_{\text{Pcon}}$ loss proposed in SteerMusic+, where in this

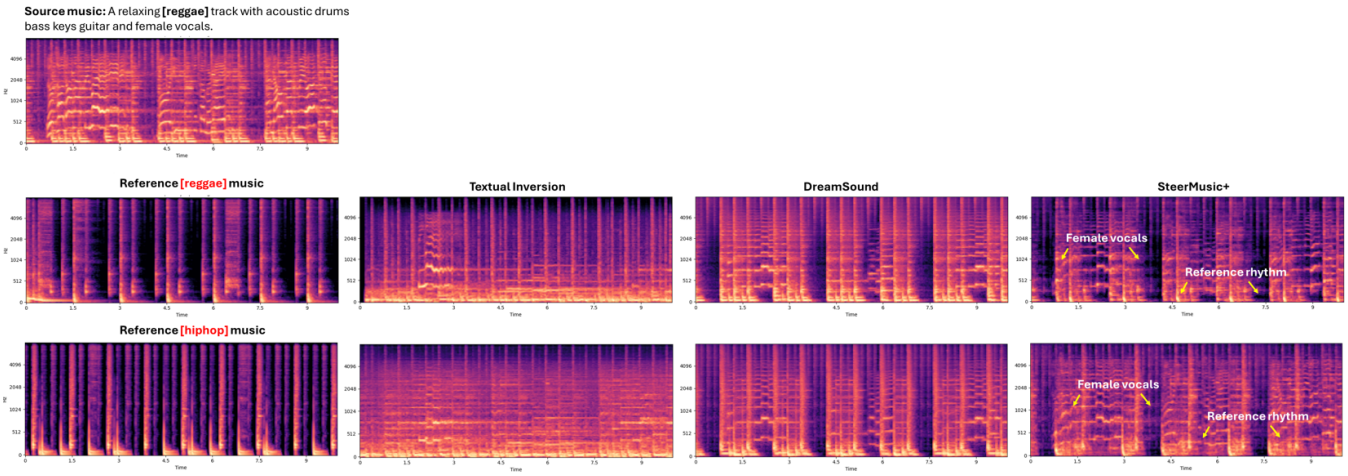

Figure 9: More visualization comparison between SteerMusic+ and baseline methods on personalized genre transfer. SteerMusic+ successfully preserve the vocal content in the source music while perform precise personalized genre transfer.

Table 2: Model comparison on personalized music genre transfer (SteerMusic+ uses the same personalized model as DreamSound)

Method	Concept	$\text{FAD}_{\text{CLAP}} \downarrow$	$\text{FAD}_{\text{Viggish}} \downarrow$	CQT-1 PCC ↑	LPAPS ↓	CDPAM ↓
Textual inv.	Morricone	0.496	2.149	0.253	4.815	0.609
DreamSound	Morricone	0.720	4.309	0.289	5.093	0.469
SteerMusic+	Morricone	**0.312**	**0.518**	**0.459**	**3.896**	**0.465**
Textual inv.	Reggae	0.446	1.928	0.199	5.062	0.804
DreamSound	Reggae	0.657	3.276	0.312	5.309	**0.700**
SteerMusic+	Reggae	**0.432**	**0.801**	**0.319**	**4.416**	0.705
Textual inv.	Sarabande	0.466	1.778	0.251	5.079	0.815
DreamSound	Sarabande	0.814	3.732	0.265	5.070	0.616
SteerMusic+	Sarabande	**0.333**	**0.398**	**0.398**	**3.997**	**0.573**
Textual inv.	Hiphop	2.868	1.545	0.293	4.607	0.832
DreamSound	Hiphop	2.280	4.288	0.258	5.209	0.702
SteerMusic+	Hiphop	**2.078**	**0.553**	**0.389**	**4.139**	**0.701**

setting, we calculate contrastive loss between two self-attention features come from the same diffusion model with respect to the spatial location. This additional loss serves the same function as the method proposed by [12], which helps to enhance the source music structure consistency during editing. Since we used a spectrogram-based text-to-audio diffusion model, the source music structture consistency here represents the structure consistency in Mel-spectrogram.

5.3 Experimental Results

We make comparison between SteerMusic and the proposed two additional adaptations in above subsections. Table 3 presents a performance comparison between the original SteerMusic method proposed in the main text and variants, denoted as SteerMusic$^{\diamond}$ and SDS. SteerMusic$^{\diamond}$ incorporates an additional contrastive loss introduced by [12] to further enhance melody preservation in the source music.

Although SDS achieves the highest CLAP score compared to SteerMusic and SteerMusic$^{\diamond}$, its significantly lower CQT-1 PCC and LPAPS scores indicate a failure to preserve source music consistency. This result consists to the finding in image editing domain [6], which SDS suffers blurry issue and make the edited results difficult to preserve original content. Additionally, SDS yields significantly higher FAD scores, further indicating lower audio quality in the edited results.

In SteerMusic$^{\diamond}$, the inclusion of the $\mathcal{L}_{\text{PatchNCE}}(x, x^{src})$ loss helps maintain the structural characteristics of the source music in the edited outputs, as evidenced by a higher CQT-1 PCC score and lower LPAPS score. However, this comes at the cost of a reduced CLAP score, suggesting that the edited outputs may be less aligned with the target prompt. This implies that SteerMusic$^{\diamond}$ produces less perceptible edits, leaning the outputs closer to the original music. These results indicate a failed adaptation of the Contrastive Denoising Score (CDS) [12], originally proposed for the image domain, to the music editing task. One possible explanation is that enforcing stronger structural consistency in the Mel-spectrogram constrains frequency-domain edits, leading to reduced editing accuracy. Enforcing structural consistency like $\mathcal{L}_{\text{PatchNCE}}(x, x^{src})$ further push

Table 3: Model comparison between SteerMusic and other score distillation adaptation methods on different music style transfer sub-tasks. SteerMusic$^{\diamond}$ represents the results with extra $\mathcal{L}_{\text{PatchNCE}}(x, x^{src})$ defined in Eq. 12 in the SteerMusic.

Method	Task	$\text{FAD}_{\text{CLAP}} \downarrow$	$\text{FAD}_{\text{Viggish}} \downarrow$	CQT-1 PCC↑	CLAP ↑	LPAPS ↓
SDS	Change instrument	2.178	1.821	0.294	0.267	4.938
SteerMusic	Change instrument	**0.257**	**0.313**	0.429	**0.269**	4.291
SteerMusic$^{\diamond}$	Change instrument	0.277	0.432	**0.685**	0.236	**3.435**
SDS	Change genre	2.529	2.476	0.233	**0.268**	5.028
SteerMusic	Change genre	0.278	**0.397**	0.439	0.249	4.013
SteerMusic$^{\diamond}$	Change genre	**0.259**	0.551	**0.647**	0.221	**3.474**
SDS	Change mood	2.801	1.764	0.284	**0.277**	4.784
SteerMusic	Change mood	0.275	0.315	0.521	0.273	**3.145**
SteerMusic$^{\diamond}$	Change mood	**0.273**	**0.313**	**0.644**	0.272	3.396
SDS	Change background	2.152	2.122	0.273	**0.268**	4.877
SteerMusic	Change background	0.312	**0.521**	0.564	0.243	3.402
SteerMusic$^{\diamond}$	Change background	**0.310**	0.832	**0.702**	0.242	**3.388**
SDS	Overall	2.410	2.061	0.270	**0.270**	4.918
SteerMusic	Overall	**0.278**	**0.381**	0.480	0.259	3.772
SteerMusic$^{\diamond}$	Overall	**0.278**	0.524	**0.669**	0.241	**3.428**

the edited output too close to the source music, suppressing necessary changes in frequency domain, such as timbre and rhythm, that are essential for aligning with the target prompt for style transfer editing.

Compared to both adaptations, SteerMusic achieves a better balance between source music consistency and edit fidelity, demonstrating its effectiveness in the music editing domain.

REFERENCES

[1] Yuval Alaluf, Daniel Garibi, Or Patashnik, Hadar Averbuch-Elor, and Daniel Cohen-Or. 2024. Cross-image attention for zero-shot appearance transfer. In *ACM SIGGRAPH 2024 Conference Papers*. 1–12.

[2] Ke Chen, Xingjian Du, Bilei Zhu, Zejun Ma, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. 2022. HTS-AT: A hierarchical token-semantic audio transformer for sound classification and detection. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 646–650.

[3] Ondřej Cífka, Alexey Ozerov, Umut Şimşekli, and Gael Richard. 2021. Self-supervised vq-vae for one-shot music style transfer. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 96–100.

[4] Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez. 2023. Simple and controllable music generation. *Advances in Neural Information Processing Systems* 36 (2023), 47704–47720.

[5] Azalea Gui, Hannes Gamper, Sebastian Braun, and Dimitra Emmanouilidou. 2024. Adapting frechet audio distance for generative music evaluation. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 1331–1335.

[6] Amir Hertz, Kfir Aberman, and Daniel Cohen-Or. 2023. Delta denoising score. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2328–2337.

[7] Jonathan Ho and Tim Salimans. [n. d.]. Classifier-Free Diffusion Guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*.

[8] Siyuan Hou, Shansong Liu, Ruibin Yuan, Wei Xue, Ying Shan, Mangsuo Zhao, and Chao Zhang. 2025. Editing Music with Melody and Text: Using ControlNet for Diffusion Transformer. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 1–5. https://doi.org/10.1109/ICASSP49660.2025.10890309

[9] Huadai Liu, Jialei Wang, Xiangtai Li, Rongjie Huang, Yang Liu, Jiayang Xu, and Zhou Zhao. 2024. MEDIC: Zero-shot Music Editing with Disentangled Inversion Control. *arXiv preprint arXiv:2407.13220* (2024).

[10] Hila Manor and Tomer Michaeli. 2024. Zero-Shot Unsupervised and Text-Based Audio Editing Using DDPM Inversion. *ICML* (2024).

[11] Chenlin Meng, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. 2021. Sdedit: Image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073* (2021).

[12] Hyelin Nam, Gihyun Kwon, Geon Yeong Park, and Jong Chul Ye. 2024. Contrastive denoising score for text-guided latent diffusion image editing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9192–9201.

[13] Manos Plitsis, Theodoros Kouzelis, Georgios Paraskevopoulos, Vassilis Katsouros, and Yannis Panagakis. 2024. Investigating personalization methods in text to music generation. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 1081–1085.

[14] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. 2022. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988* (2022).

[15] Jiaming Song, Chenlin Meng, and Stefano Ermon. [n. d.]. Denoising Diffusion Implicit Models. In *International Conference on Learning Representations*.

[16] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. 2023. Plug-and-play diffusion features for text-driven image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1921–1930.

[17] Yusong Wu, Ke Chen, Tianyu Zhang, Yuchen Hui, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. 2023. 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)*. IEEE, 1–5.

[18] Zhengyang Yu, Zhaoyuan Yang, and Jing Zhang. 2025. DreamSteerer: Enhancing Source Image Conditioned Editability using Personalized Diffusion Models. *Advances in Neural Information Processing Systems* 37 (2025), 120699–120734.

[19] Junyi Zhang, Charles Herrmann, Junhwa Hur, Luisa Polania Cabrera, Varun Jampani, Deqing Sun, and Ming-Hsuan Yang. 2023. A tale of two features: Stable diffusion complements dino for zero-shot semantic correspondence. *Advances in Neural Information Processing Systems* 36 (2023), 45533–45547.

[20] Yixiao Zhang, Yukara Ikemiya, Gus Xia, Naoki Murata, Marco A Martínez-Ramírez, Wei-Hsiang Liao, Yuki Mitsufuji, and Simon Dixon. 2024. MusicMagus: zero-shot text-to-music editing via diffusion models. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*. 7805–7813.