

EG-VAE: A Unified Framework for Electric Guitar Tone Transfer and Removal

Yen-Tung Yeh¹, Yun-Ning (Amy) Hung², Yi-Hsuan Yang¹

¹ National Taiwan University ² Music AI

Abstract—Electric guitar tone transfer (EGTT) and tone removal (EGTR) are two fundamental highly related tasks in guitar tone modeling: EGTT replaces a recording’s tone with that of a reference, while EGTR recovers the dry direct-input (DI) signal from a wet processed recording. Despite their highly related nature, prior work has addressed them independently, and both works has yet to achieve satisfactory results. In this paper, we propose EG-VAE, a unified framework that jointly models EGTT and EGTR by disentangling frame-level content and global tone representations from wet recordings with a variational autoencoder. Transfer recombines a source’s content with a reference’s tone, while removal is realized by a novel tone masking objective that serves a dual role: enforcing content–tone disentanglement during training and realizing the removal procedure at inference. To improve transfer to tones unseen in training, a second training stage shapes a smooth tone space through variational sampling and audio-effects augmentation. Experimental results from both objective and subjective evaluations demonstrate that EG-VAE outperforms task-specific baselines on transfer and remains competitive with models on removal. Demos are available at [URL].

Index Terms—Electric guitar tone transfer, tone removal, audio effects, guitar disentanglement

I. INTRODUCTION

The electric guitar has been a long-standing subject of study in the music information retrieval (MIR) and audio effects communities, across tasks such as transcription [1]–[5], performance rendering [6], and pedal/amplifier modeling [7]–[10]. In a typical electric guitar recording, the *dry* direct-input (DI) signal is processed by a chain of effects pedals and an amplifier, producing the *wet* signal heard by the listener [1]. The character imparted by this chain is the recording’s *tone*, whose variation has direct consequences across these tasks: transcription accuracy, for instance, degrades substantially under heavily processed tones [2]. We study two underexplored problems focusing on tone itself: *electric guitar tone transfer* (EGTT), which replaces a recording’s tone with that of another, and *electric guitar tone removal* (EGTR), which recovers the dry DI signal from a wet recording. Figure 1 illustrates the two tasks and their relationship to the underlying signal chain.

Although singing voice [11] and general music [12]–[14] have shown promising results in audio effects style transfer, EGTT remains challenging [12], owing to the highly nonlinear distortion in its signal chain and the wide diversity of tones it can produce, spanning all four effect categories of Section II-A (distortion, modulation, equalization, and spatial). We observe

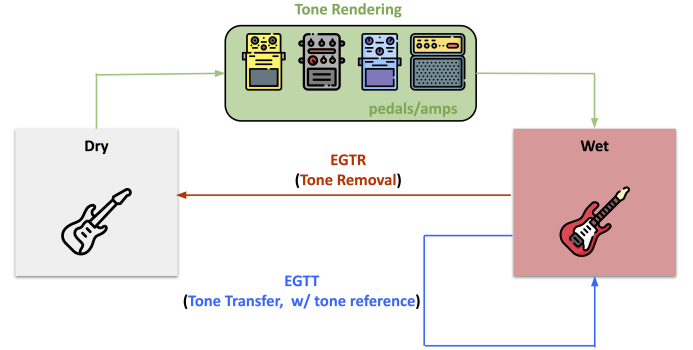


Fig. 1. The two tone-modeling tasks studied in this work. A dry direct-input (DI) signal is rendered into a wet recording by a chain of effects pedals and an amplifier (top, green). Tone removal (EGTR, red) recovers the dry signal from the wet recording, inverting the rendering process. Tone transfer (EGTT, blue) replaces the tone of the wet recording with that of another wet reference, preserving the played content.

that the common formulation of effects transfer is not well suited to EGTT. These methods estimate a tone from a reference and apply it to the input, implicitly assuming the input is *dry* or only lightly processed. This assumption fails for guitar, whose input is a *wet* recording carrying a chain of non-invertible effects. A natural route to faithful transfer is thus to first remove the input’s own tone, making removal a prerequisite for transfer rather than a separate task. Prior approaches that sidestep this issue, which assumes a dry DI input and targets only the amplifier, are restricted cases of EGTT, addressing neither the wet-input setting nor the full signal chain [15].

This recovery of the dry DI signal from a wet recording is precisely the task of EGTR. Existing removal methods, however, still struggle to produce high-quality audio. Prior work frames EGTR as signal enhancement [16]–[18] and addresses only restricted settings: synthetic, simplified effects such as hard clipping [16], a narrow range of effect types that excludes reverb and modulation [17], or general audio rather than guitar [18]. None removes the *full* guitar chain, by which we mean coverage of all common guitar pedal and amplifier types. More fundamentally, each treats removal as a standalone problem, producing a representation specialized to removal and disconnected from transfer.

We argue that the limitations of both tasks share a single cause: each is formulated in isolation, overlooking that the two act on the same attribute, the recording’s tone. Transfer replaces this tone with another, while removal strips it away to

recover the dry signal. The two are therefore complementary, manipulating one shared object in opposite directions. This naturally raises the question: can unifying EGTT and EGTR via representation learning overcome the limitations of task-specific methods?

We propose EG-VAE, which factorizes a wet recording into a content embedding and a tone embedding through a variational autoencoder. Three challenges remain. First, the factorization is non-trivial to obtain, as tone information tends to leak into the content embedding, a persistent failure mode in disentangled representation learning [19]. Second, EGTR has no reference to specify its target: it must recover the dry signal from the wet input alone. We address both with the proposed *tone masking*, which removes the tone from the decoder’s input, leaving the content embedding to reconstruct the signal alone. This single mechanism serves two roles: it defines EGTR, as the masked forward pass directly produces the dry DI signal; and it suppresses leakage, since the content embedding, forced to reconstruct the dry signal without the tone, must learn to exclude tone information. The third challenge is generalization: at inference, EGTT must handle reference tones unseen during training. We address this in a second training stage with two complementary mechanisms: variational sampling under KL regularization, which shapes a smooth tone space so that tones near the training distribution decode coherently, and audio-effects augmentation, which broadens that distribution to better cover unseen tones.

We evaluate EG-VAE against EGTT and EGTR baselines in two settings: seen tones, whose signal chains appear in training, and unseen tones, which assess generalization to chains held out from training. A single unified model handles both tasks: EG-VAE reduces the Mel spectral distance of transfer by 44% over the strongest task-specific baseline (One-to-many with EGTR enhancement [15], 0.86 vs. 1.53 on seen tones) and improves over the best removal model (Distortion Recovery [17], 1.10 vs. 1.21 Mel on seen tones, matching on unseen). An ablation study further confirms the contribution of each disentanglement mechanism and of the second-stage design on unseen tones.

Our main contributions are summarized as follows:

- We propose EG-VAE, to our knowledge the first unified framework for EGTT and EGTR, learning a single factorization of a wet recording into content and tone that supports both tasks.
- We introduce tone masking, a single objective shared by training and inference that defines the EGTR procedure while reinforcing content–tone disentanglement.
- We design a two-stage training strategy that smooths the tone space through variational sampling and audio-effects augmentation, improving transfer to unseen tones.

This paper is organized as follows. Section II reviews the electric guitar signal chain and prior work on EGTT and EGTR. Section III presents EG-VAE, covering its problem formulation, architecture, and two-stage training procedure. Section IV describes the experimental setup, including the dataset, baselines, and evaluation protocol. Section V reports objective and subjective results together with ablation studies, and Section ?? concludes.

II. BACKGROUND AND RELATED WORK

A. The Electric Guitar Signal Chain

An electric guitar produces sound through magnetic pickups that convert string vibration into an electrical signal, the *direct-input (DI)* signal or *dry signal*, denoted $\mathbf{x}_{\text{dry}} \in \mathbb{R}^T$ for a recording of T samples. To produce the sound delivered to the listener, the dry signal is routed through a *signal chain* consisting of one or more audio effects processors followed by an amplifier, yielding the *wet signal* $\mathbf{x}_{\text{wet}} \in \mathbb{R}^T$:

$$\mathbf{x}_{\text{wet}} = f(\mathbf{x}_{\text{dry}}) = (f_{\text{amp}} \circ f_{\text{fx}})(\mathbf{x}_{\text{dry}}), \quad (1)$$

where $f : \mathbb{R}^T \rightarrow \mathbb{R}^T$ is the transformation applied by the signal chain, f_{fx} denotes the composition of effects processors preceding the amplifier, and f_{amp} denotes the amplifier.¹ We refer to f as the recording’s *tone*. The effects sub-chain f_{fx} is itself a composition of M processors applied in sequence,

$$f_{\text{fx}} = f_M \circ f_{M-1} \circ \cdots \circ f_1, \quad (2)$$

where each f_i corresponds to a single audio effects processor and f_1 is applied first. We note that each f_i may be time-varying and signal-dependent, with internal parameters omitted from the notation for clarity. Audio effects processors span a broad design space; commonly recognized categories include *distortion* (non-linear, e.g., overdrive, fuzz), *modulation* (time-varying, e.g., chorus, flanger), *equalization* (linear spectral shaping), and *spatial* (convolutional, e.g., reverb), though many processors combine these or fall outside any single category. The amplifier f_{amp} , in turn, is itself a multi-stage processor, typically comprising a preamplifier, tone stack, power amplifier, and speaker cabinet, each contributing its own non-linear or filtering behavior [10]. Even within a recognized category, different implementations produce perceptually distinct sounds owing to differences in circuit topology and analog non-linearities [20], [21]. Combined with the combinatorial structure of both f_{fx} and f_{amp} , this variation makes tone a complex, high-dimensional function.

B. Electric Guitar Tone Transfer

Direct work on EGTT is limited. The closest prior work treats EGTT as a one-to-many audio effects modeling problem. Chen *et al.* [15] use contrastive-pretrained tone embeddings for amplifier modeling, training a neural network on paired (dry, wet) data to emulate multiple amplifiers conditioned on a reference recording. In the notation of Section II-A, this targets f_{amp} , while assuming the input is the dry DI signal $\mathbf{x}_{\text{dry}}$ with no effects chain f_{fx} . The broader audio effects style transfer literature instead targets f_{fx} , and treats it more flexibly. Steinmetz *et al.* [14] use differentiable signal processing [22] to estimate f_{fx} parameters end-to-end, requiring the order and type of each f_i to be specified before training (e.g., parametric equalizer $\rightarrow$ compressor in their work); this constrains the model to a fixed configuration and limits its flexibility on signal chains that do not match the predefined structure. Subsequent work [11], [13]

¹We adopt the conventional signal flow in which the effects sub-chain precedes the amplifier. In practice, some effects may be placed after the amplifier, reverberation, for instance, but we omit such cases from the notation for simplicity, as they do not affect the formulation that follows.

relaxes this training-time requirement through inference-time optimization, but the chain structure itself remains fixed. ST-ITO [12] further generalizes this line by lifting the predefined-chain requirement, supporting arbitrary f_{fx} at inference.

Yet even with this flexibility, a more fundamental limitation remains, concerning the input itself rather than the chain structure. These methods impose no formal restriction on the input signal, but they ultimately produce an output of the form

$$\hat{\mathbf{x}} = \hat{f}_{\text{fx}}(\mathbf{x}_{\text{base}}), \quad (3)$$

where the estimated chain $\hat{f}_{\text{fx}}$ is applied to a base signal $\mathbf{x}_{\text{base}}$. This works well when $\mathbf{x}_{\text{base}}$ is dry or only lightly processed. In the setting we target, however, the input is already a wet recording, $\mathbf{x}_{\text{base}} = f(\mathbf{x}_{\text{dry}})$, carrying an unknown chain f . Applying a new chain on top then yields $\hat{f}_{\text{fx}} \circ f$: the reference tone is stacked onto the existing one rather than replacing it. Faithful transfer would first require removing f , precisely the tone-removal problem (EGTR), which these methods do not address. This difficulty is reflected in ST-ITO [12], whose authors report guitar tone matching as a particularly challenging case.

Across these methods, EGTT thus remains an open problem. Prior work addresses either f_{amp} [15] or f_{fx} [12], [14] in isolation, and none has been demonstrated in the setting, where the input already carries an unknown chain that must be removed before a new tone can be applied.

C. Electric Guitar Tone Removal

Whereas tone transfer replaces one tone with another, tone removal seeks to undo the signal chain entirely: given a wet recording $\mathbf{x}_{\text{wet}} = f(\mathbf{x}_{\text{dry}})$, the goal is to recover the dry DI signal,

$$\hat{\mathbf{x}}_{\text{dry}} = f^{-1}(\mathbf{x}_{\text{wet}}) \approx \mathbf{x}_{\text{dry}}, \quad (4)$$

i.e., to invert the unknown chain f . This inversion is fundamentally ill-posed: f is generally non-invertible, as nonlinear stages such as distortion clip and saturate the signal, discarding information that cannot be analytically restored. Removal must therefore be *learned* from data rather than computed in closed form, and—unlike transfer, which derives its target tone from a reference recording—removal has no reference: the dry target must be produced from the wet input alone.

Existing work has approached EGTR along two lines, both framing removal as a signal enhancement or specialized source separation problem [16]–[18]. Guitar-specific approaches focus on distortion-related effects: Imort *et al.* [16] formulate distortion removal as source separation, and Lee *et al.* [17] use a two-stage Mel-denoiser and HiFi-GAN vocoder [23] to remove a wider set of effects. In the notation of Section II-A, these methods invert f_{amp} together with the components of f_{fx} that exclude modulation and spatial processing. A separate line targets general-purpose effect removal: RemFX [18] covers a broader effect set including reverb and chorus, but uses architectures developed for general audio and is not specialized to the structure of f_{amp} or to guitar-specific instances of f_{fx} .

Two limitations follow from this framing. First, no single method inverts the *full* guitar chain: guitar-specific approaches omit the modulation and spatial components of f_{fx} , while

methods that cover these are not specialized to guitar. Second, and more fundamentally, treating removal as a standalone enhancement task yields a representation usable only for removal, disconnected from transfer even though both operate on the same function f . The possibility of modeling EGTR and EGTT jointly thus remains unexplored, despite their shared structure suggesting the two tasks could benefit each other.

D. Transfer Tasks in the Audio Domain

Beyond the task-specific prior work discussed above, three transfer tasks in the audio domain share methodological similarities with our setting but differ from guitar tone modeling in structurally important ways.

Voice conversion factorizes speech into linguistic content and speaker identity [24]–[27], modifying the latter while preserving the former. Its disentangled factor, speaker identity, is a property of the physical *source*, the vocal tract and articulatory habits of the speaker, and is bounded by what a human vocal tract can produce. Guitar tone is structurally different: the DI signal is fixed, and tone arises from the *post-source* signal chain, spanning the non-linear, time-varying, and convolutional transformations of an arbitrary chain.

Instrument timbre transfer factorizes audio into musical content and instrument identity, typically converting between distinct instrument categories such as violin and flute [22], [28], [29]. The disentangled factor is the instrument label itself: each instrument is treated as a single point in timbre space, ignoring how the timbre of single instrument can vary. However, for electric guitar, this variation is exactly what we aim to model: one guitar can produce a wide range of tones, determined not by the instrument body but by its signal chain. A categorical, between-instrument formulation cannot capture this, as it operates at a coarser level of granularity than guitar tone requires.

These two tasks fall short of guitar tone modeling for distinct reasons: voice conversion’s disentangled factor is a bounded source property rather than a post-source transformation; and instrument timbre transfer captures only categorical, between-instrument identity rather than continuous within-instrument variation.

E. Disentangled Sequential Autoencoders

A disentangled sequential autoencoder (DSAE) is a generative model that factorizes an observed time sequence into two latent variables capturing distinct aspects of the data [30]. Given an input sequence $\mathbf{x}_{1:T}$ of length T , a sequence of frame-level latent variables $\mathbf{z}_{1:T}$ captures local, time-varying features, while a latent vector $\mathbf{s}$ captures global, time-invariant factors of variation. Under the model parameters θ , the joint distribution factorizes as

$$p_{\theta}(\mathbf{x}_{1:T}, \mathbf{z}_{1:T}, \mathbf{s}) = \prod_{t=1}^T p_{\theta}(\mathbf{x}_t | \mathbf{s}, \mathbf{z}_t) p_{\theta}(\mathbf{z}_t) p_{\theta}(\mathbf{s}), \quad (5)$$

where the priors on $\mathbf{z}_t$ and $\mathbf{s}$ are independent, encouraging disentanglement between the local and global factors. The model is typically learned within the variational autoencoder

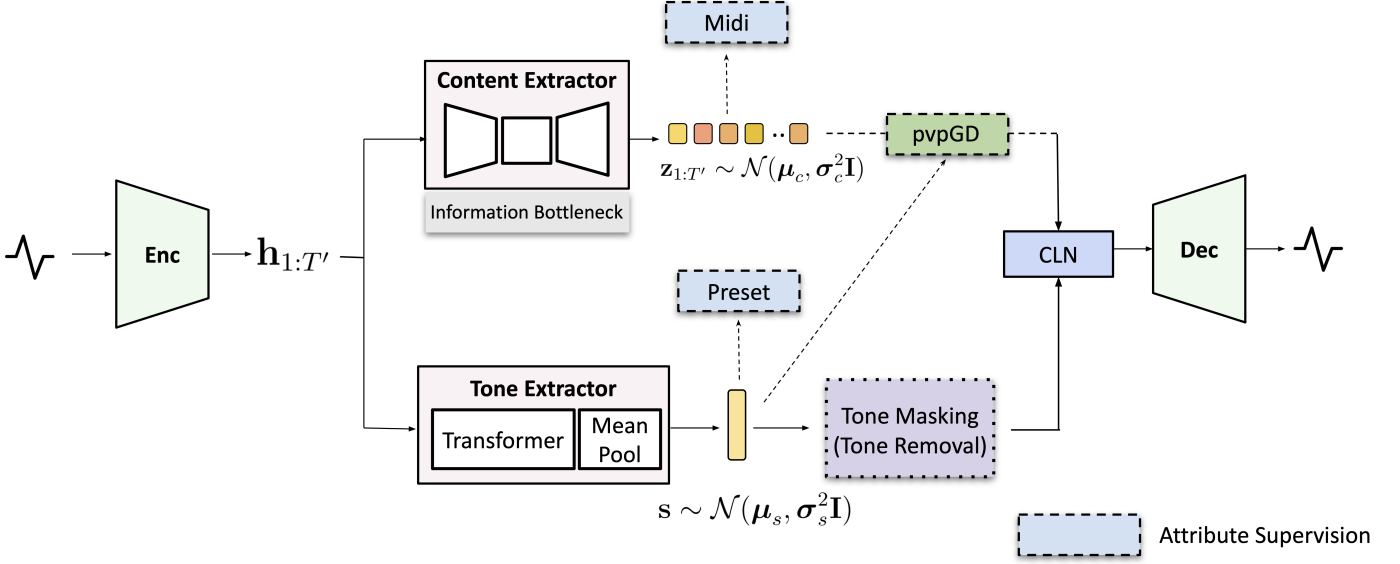


Fig. 2. Overview of EG-VAE. A shared encoder maps the input waveform to features $\mathbf{h}_{1:T'}$, feeding a content extractor that produces a frame-level latent $\mathbf{z}_{1:T'} \sim \mathcal{N}(\mu_c, \sigma_c^2 \mathbf{I})$ under an information bottleneck, and a tone extractor that produces a global tone embedding $\mathbf{s} \sim \mathcal{N}(\mu_s, \sigma_s^2 \mathbf{I})$. The two are combined by conditional layer normalization (CLN) and decoded into the output. Dashed boxes are attribute-supervision heads; pvpGD perturbs $\mathbf{z}_{1:T'}$ to prevent collapse of $\mathbf{s}$. Tone masking overrides the CLN modulation ($\gamma=1, \beta=0$) and realizes tone removal. Sampling and pvpGD are active only in stage 2.

(VAE) framework [31] by maximizing the evidence lower bound on $\log p_\theta(\mathbf{x}_{1:T})$, with KL regularization controlling the information capacity of $\mathbf{s}$ as in β -VAE [32].

This latent structure aligns naturally with the guitar signal chain of Section II-A: the frame-level $\mathbf{z}_{1:T}$ corresponds to the time-varying content of the dry DI signal, while the global $\mathbf{s}$ corresponds to the time-invariant tone f . However, this alignment alone is not sufficient for our setting. First, prior independence does not guarantee posterior disentanglement: in practice, the global factor is often absorbed by the temporal mean of the frame-level latent, allowing the decoder to ignore $\mathbf{s}$ and causing tone information to leak into $\mathbf{z}_{1:T}$ [19]. Second, the DSAE defines only a generative factorization; it provides no mechanism for tone removal, which requires a designated operating point of $\mathbf{s}$ that the decoder maps to the dry signal. These two gaps motivate the mechanisms introduced in Section III.

III. EG-VAE

We present EG-VAE, a unified model for electric guitar tone transfer (EGTT) and tone removal (EGTR). The model is trained in two stages: stage 1 establishes a disentangled factorization of wet recordings into content and tone embeddings, and stage 2 introduces variational sampling on the tone embedding together with audio-effects augmentation to improve unseen tone generalizability. We describe the factorization (Section III-A), the architecture (Section III-B), the training objectives (Sections III-C, III-D, and III-E), and the training strategy that organizes them (Section III-F).

A. Problem Formulation

We aim to develop a unified model that performs both EGTT and EGTR on wet electric guitar recordings. A wet recording

$\mathbf{x}_{\text{wet}} \in \mathbb{R}^T$ is produced by applying a signal chain f to a dry DI signal $\mathbf{x}_{\text{dry}} \in \mathbb{R}^T$, and we refer to f as the recording's tone. We learn a representation that factorizes $\mathbf{x}_{\text{wet}}$ into two disentangled attributes: a frame-level *content embedding* $\mathbf{z}_{1:T'}$ and a global *tone embedding* $\mathbf{s}$. The content embedding $\mathbf{z}_{1:T'}$, of length $T' < T$, operates at a temporal resolution downsampled from the input by the encoder and captures the time-varying content of the dry DI signal, including the playing style; the tone embedding $\mathbf{s}$ captures the signal chain f . Ideally, $\mathbf{s}$ should encode the identity, ordering, and parameters of the audio effects processors composing f , while remaining invariant to the musical content. The model consists of an encoder $\mathcal{E}$ producing $(\mathbf{z}_{1:T'}, \mathbf{s}) = \mathcal{E}(\mathbf{x}_{\text{wet}})$ and a decoder $\mathcal{D}$ producing $\hat{\mathbf{x}} = \mathcal{D}(\mathbf{z}_{1:T'}, \mathbf{s})$. Both tasks are then realized by manipulating the tone embedding $\mathbf{s}$ while preserving the content embedding. For EGTT, $\mathbf{s}$ is replaced with the tone embedding $\mathbf{s}^{\text{ref}}$ of a reference recording, yielding

$$\hat{\mathbf{x}}^{\text{EGTT}} = \mathcal{D}(\mathbf{z}_{1:T'}^{\text{src}}, \mathbf{s}^{\text{ref}}), \quad (6)$$

whose content and reference need not share musical content. For EGTR, the tone is instead removed rather than replaced, mapping the input to its dry DI signal, $\hat{\mathbf{x}}^{\text{EGTR}} \approx \mathbf{x}_{\text{dry}}$; we develop this masking operation in Section III-D.

B. Model Architecture

EG-VAE adapts the DAC-VAE [33], the VAE-version of DAC codec [34], for disentangled representation learning. As shown in Figure 2, a shared encoder maps the wet input to a frame-level latent, from which two extractors produce the embeddings $\mathbf{z}_{1:T'}$ and $\mathbf{s}$. A conditional layer normalization (CLN) module [35] then combines them, and a decoder reconstructs the audio waveform.

a) *Shared encoder and decoder*: Rather than using two separate encoders for the two factors [24], [36], we use a single shared encoder. Both content and tone are derived from the same underlying acoustic features, leaving the attribute-specific factorization to the two lightweight extractors that follow. The encoder is a stack of 1D convolutional residual blocks with Snake activations [37] and progressive temporal downsampling, mapping the wet input $\mathbf{x}_{\text{wet}} \in \mathbb{R}^T$ to a frame-level latent $\mathbf{h} \in \mathbb{R}^{D \times T'}$, where D is the channel dimension and T' is the downsampled temporal resolution. The decoder mirrors this structure with progressive temporal upsampling, reconstructing the waveform from the CLN-combined representation $\tilde{\mathbf{z}}_{1:T'}$.

b) *Content and tone extractors*: The content extractor produces $\mathbf{z}_{1:T'}$ through a per-frame variational bottleneck. A 1×1 convolution projects $\mathbf{h}$ to per-frame posterior parameters $(\boldsymbol{\mu}_c, \boldsymbol{\sigma}_c)$ of a diagonal Gaussian, from which $\mathbf{z}_t \sim \mathcal{N}(\boldsymbol{\mu}_{c,t}, \boldsymbol{\sigma}_{c,t}^2 \mathbf{I})$ is sampled; a second 1×1 convolution projects the sampled embedding back to the latent dimensionality. The tone extractor produces the global $\mathbf{s}$ in two steps: a transformer encoder processes $\mathbf{h}$ to capture dependencies across the temporal dimension, and a temporal pooling operation aggregates the result into a single fixed-dimensional vector. A linear variational head then produces the global posterior parameters $(\boldsymbol{\mu}_s, \boldsymbol{\sigma}_s)$, from which $\mathbf{s} \sim \mathcal{N}(\boldsymbol{\mu}_s, \boldsymbol{\sigma}_s^2 \mathbf{I})$ is sampled.

c) *Embedding combination via CLN*: The CLN module combines the two embeddings as

$$\tilde{\mathbf{z}}_t = \gamma(\mathbf{s}) \odot \text{LayerNorm}(\mathbf{z}_t) + \beta(\mathbf{s}), \quad (7)$$

where $\gamma(\mathbf{s})$ and $\beta(\mathbf{s})$ are produced by a linear projection of $\mathbf{s}$, and $\text{LayerNorm}(\cdot)$ [38] omits learnable affine parameters. The combined $\tilde{\mathbf{z}}_{1:T'}$ is then decoded into the waveform.

C. Disentanglement Mechanisms

The architecture provides a factorization into $\mathbf{z}_{1:T'}$ and $\mathbf{s}$, but does not guarantee disentanglement directly. The core difficulty is that the frame-level latent $\mathbf{z}_{1:T'}$ is expressive enough to encode the global tone factor across its frames, letting the decoder reconstruct the wet signal while ignoring $\mathbf{s}$. Our design therefore limits the capacity of $\mathbf{z}_{1:T'}$ so that the global factor is forced into $\mathbf{s}$. We apply four mechanisms during training: an information bottleneck on $\mathbf{z}_{1:T'}$ limits its per-frame capacity; content–tone perturbation forces each embedding to encode only its intended factor by recombining content and tone drawn from different recordings; attribute supervision provides external targets for what each embedding should encode; and posterior variance-parameterised Gaussian dropout (pvpGD) [19] prevents posterior collapse of the tone latent. A fifth mechanism, tone masking, is the key to unifying the two tasks and is introduced separately in Section III-D.

a) *Information bottleneck on $\mathbf{z}_{1:T'}$* : Inspired by [24], we apply a KL regularization on $\mathbf{z}_{1:T'}$ against a standard normal prior to constrain its capacity:

$$\mathcal{L}_{\text{KL}}^c = \text{KL}(q(\mathbf{z}_{1:T'} | \mathbf{x}_{\text{wet}}) \| p(\mathbf{z})), \quad p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I}). \quad (8)$$

The regularization is applied in a lower-dimensional space: the encoder features are first projected from 1024 to 128 dimensions, where the KL term is computed, and then projected back

to the original dimensionality to form the content embedding. Weighted by λ_{KL}^c , this term limits the information capacity of $\mathbf{z}_{1:T'}$, as in β -VAE [32], pressuring it to encode only the information necessary for reconstruction.

b) *Content–tone perturbation*: Inspired by [39], this mechanism forces each embedding to be invariant to the factor it should not encode. Let $\mathbf{x}_{\text{wet}}^{(c,t)}$ denote a wet recording with content identifier c (the dry DI signal it is rendered from) and tone identifier t (the preset applied to that DI signal). At each training step we sample a content-match $\mathbf{x}_{\text{wet}}^{(c_1, t_0)}$, a tone-match $\mathbf{x}_{\text{wet}}^{(c_0, t_0)}$, and a target $\mathbf{x}_{\text{wet}}^{(c_0, t_1)}$. We extract the content embedding from the content-match, the tone embedding from the tone-match, and supervise the decoder to reconstruct the target:

$$\mathcal{L}_{\text{rec}}^{\text{c-t}} = \mathcal{L}_{\text{audio}}(\mathcal{D}(\mathbf{z}_{1:T'}^{(c_0, t_1)}, \mathbf{s}^{(c_1, t_0)}), \mathbf{x}_{\text{wet}}^{(c_0, t_0)}). \quad (9)$$

Neither input matches the target on both factors, since the content-match supplies content c_0 with a different tone t_1 , while the tone-match supplies tone t_0 with a different content c_1 . Reconstructing $\mathbf{x}_{\text{wet}}^{(c_0, t_0)}$ therefore requires $\mathbf{z}_{1:T'}^{(c_0, t_1)}$ to encode only c_0 and $\mathbf{s}^{(c_1, t_0)}$ to encode only t_0 ; any residual leakage in either direction hurts the reconstruction.

c) *Attribute supervision*: Inspired by [25], we supervise each embedding with an auxiliary classification task on the attribute it should capture. The content embedding $\mathbf{z}_{1:T'}$ is passed to a frame-level pitch classifier that predicts, for each frame, the set of MIDI pitches active at that frame. Because guitar performance is polyphonic, multiple pitches may be active simultaneously, so this is formulated as multi-label classification with binary cross-entropy applied independently per pitch ($\mathcal{L}_{\text{pitch}}$). The tone embedding $\mathbf{s}$ is passed to a global *preset* classifier, where a preset—a specific combination of effects and amplifier settings with their parameter values—is the standard unit for describing a guitar tone. Our training data is annotated with one preset label per recording, which we use as a discrete tone identifier to guide $\mathbf{s}$; this classifier is trained with categorical cross-entropy ($\mathcal{L}_{\text{preset}}$). Because preset supervision is categorical, it promotes separation between training tones but does not encourage to produce a continuous, smooth tone space; we address this in stage 2 (Section III-F).

d) *pvpGD: probabilistic perturbation guided by tone variance*: A further disentanglement difficulty, distinct from the leakage above, is posterior collapse of the global latent: under KL regularization on $\mathbf{s}$, the posterior variance can grow toward $\boldsymbol{\sigma}_s \rightarrow \mathbf{1}$, at which point $\mathbf{s}$ loses informativeness and the frame-level latent $\mathbf{z}_{1:T'}$ absorbs the modeling capacity that should reside in $\mathbf{s}$ [19]. This collapses the factorization: tone information migrates back into $\mathbf{z}_{1:T'}$, the precise leakage our other mechanisms work to prevent. To counteract this, we adopt the pvpGD mechanism of [19], which couples the capacity of $\mathbf{z}_{1:T'}$ to that of $\mathbf{s}$. Before CLN combination, the sampled frame-level latent is perturbed by multiplicative Gaussian noise whose scale is set by the tone posterior variance:

$$\mathbf{z}_t^{\text{pvp}} = \mathbf{z}_t \odot (1 + \bar{\sigma}_s \boldsymbol{\epsilon}_t), \quad \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad (10)$$

where $\bar{\sigma}_s = \exp(\frac{1}{D_s} \sum_{i=1}^{D_s} \log \sigma_{s,i})$ is the geometric mean of the per-dimension tone-posterior standard deviations, with

D_s the tone-embedding dimensionality and $\sigma_{s,i}$ the i -th component of σ_s , and $\mathbf{z}_t^{\text{pvp}}$ replaces $\mathbf{z}_t$ as the input to CLN in Eq. (7). This couples the two latents from both sides: when σ_s grows—the signature of collapse—the perturbation on $\mathbf{z}_{1:T'}$ strengthens and degrades reconstruction, penalizing collapse; conversely, shrinking σ_s to reclaim capacity for $\mathbf{z}_{1:T'}$ is penalized by $\mathcal{L}_{\text{KL}}^s$. The two pressures bound σ_s , holding $\mathbf{s}$ at a capacity that retains tone information. pvpGD is thus a disentanglement mechanism in nature: it keeps the tone latent informative so the factorization does not collapse. Because it requires a stochastic tone path, however, it is operative only once variational sampling is introduced in stage 2 (Section III-F); we therefore evaluate its effect in the stage-2 ablation (Table IV). At inference, $\bar{\sigma}_s$ is set to zero and the perturbation vanishes.

D. Tone Masking

Tone masking removes the tone from the combined representation, leaving the content embedding to reconstruct the dry DI signal alone (Figure 3). This serves two roles: a disentanglement objective during training, and the EGTR procedure at inference.

a) *The masking forward pass:* Recall that CLN modulates the content embedding through $\gamma(\mathbf{s})$ and $\beta(\mathbf{s})$ (Eq. (7)). Masking overrides this modulation directly, setting $\gamma = 1$ and $\beta = 0$ regardless of $\mathbf{s}$, so the content embedding passes through unmodulated:

$$\tilde{\mathbf{z}}_t = \text{LayerNorm}(\mathbf{z}_t). \quad (11)$$

We call this the *no-modulation regime* and write the corresponding decoding as $\mathcal{D}_{\text{mask}}(\mathbf{z}_{1:T'})$. Because the override acts directly on γ and β , it is independent of the value of $\mathbf{s}$ and of how the CLN projection is trained.

This operation has a direct interpretation in terms of normalization. Layer normalization removes a representation’s global statistics, and conditional normalization layers re-introduce a global factor through their affine parameters (γ, β) , a mechanism widely used for style transfer [40] and attribute-conditioned synthesis [35]. CLN follows this design, injecting the tone factor through $(\gamma(\mathbf{s}), \beta(\mathbf{s}))$. Setting $\gamma = 1$ and $\beta = 0$ thus reduces CLN to plain layer normalization, and tone masking acts as the inverse of tone conditioning.

b) *Training: masking as a disentanglement objective:* During training, the masked pass is supervised against the dry DI signal:

$$\mathcal{L}_{\text{mask}} = \mathcal{L}_{\text{audio}}(\mathcal{D}_{\text{mask}}(\mathbf{z}_{1:T'}), \mathbf{x}_{\text{dry}}). \quad (12)$$

Under the no-modulation regime, $\mathbf{z}_{1:T'}$ is the decoder’s only input, so it must carry everything needed to reconstruct $\mathbf{x}_{\text{dry}}$. Together with the bottleneck $\mathcal{L}_{\text{KL}}^c$, which caps its capacity, this leaves no room for tone-related information that the dry target does not require, improving the content–tone factorization. The objective also fixes the meaning of the no-modulation regime as the state the decoder maps to the dry signal, consistent with the dry targets in our data (Section III-A).

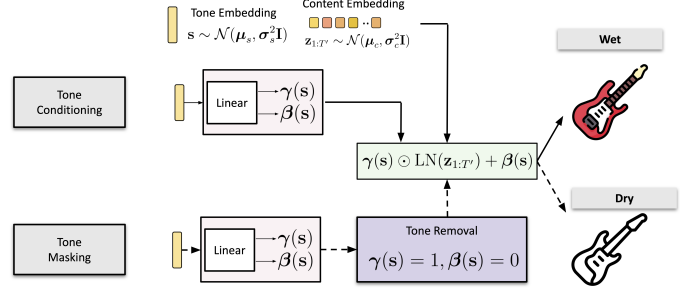


Fig. 3. Tone conditioning versus tone masking, sharing the same CLN operation $\gamma(\mathbf{s}) \odot \text{LN}(\mathbf{z}_{1:T'}) + \beta(\mathbf{s})$ from the tone embedding, yielding a wet output; masking overrides them to $\gamma = 1, \beta = 0$, leaving the content unmodulated and yielding the dry DI signal. Tone removal is thus the masked decoding path.

c) *Inference: masking as the EGTR procedure:* EGTR reuses this masked pass unchanged, $\hat{\mathbf{x}}^{\text{EGTR}} = \mathcal{D}_{\text{mask}}(\mathbf{z}_{1:T'})$. Because training and inference share an identical operation, removal needs no extra module, fine-tuning, or task-specific procedure: it is a direct readout of the representation learned for transfer, which is the precise sense in which EG-VAE unifies the two tasks.

E. Unseen Tone Generalizability

At inference, EGTT must handle *unseen* reference tones, those extracted from recordings whose signal chains are not in the training set. Generalizing to such tones requires two properties, addressed by two mechanisms in stage 2: the decoder must vary *smoothly* with $\mathbf{s}$, so that a tone near a training tone decodes coherently; and the training tones must *cover* the space densely enough that unseen tones fall near them. The first is a property of how the decoder responds to $\mathbf{s}$; the second, of how the training tones are distributed.

a) *Variational sampling on the tone path:* To make the decoder vary smoothly with $\mathbf{s}$, we treat $\mathbf{s}$ as a stochastic latent and sample $\mathbf{s} \sim \mathcal{N}(\mu_s, \sigma_s^2 \mathbf{I})$ during training, unlike prior work that uses a deterministic, unregularized timbre embedding [25], [41]. Each step exposes the decoder to a neighborhood of μ_s , forcing it to produce coherent output across that neighborhood rather than at isolated points [31]; the KL regularization $\mathcal{L}_{\text{KL}}^s$ extends this continuity globally by pulling the posterior toward a standard normal prior. A deterministic embedding offers no such guarantee, supporting neither sampling of new tones nor smooth interpolation [42], [43].

b) *Audio-effects augmentation:* Smoothness alone is not sufficient: the training tones form a discrete set, and unseen tones may fall in regions far from any of them. We therefore augment each preset’s recordings with additional effects, producing variants that sit near the preset but realize slightly different chain configurations. This densifies the distribution of training tones, so that an unseen tone is more likely to fall near data the decoder has seen. Rather than arbitrary effects, the augmentation follows the Wiener–Hammerstein model, a linear–nonlinear–linear cascade widely used to model guitar amplifiers [10]: gain and parametric equalization form

the linear stages and distortion the nonlinearity. Applying the distortion with a fixed probability lets the chain realize either a linear or a nonlinear system, covering both regimes of guitar signal chains.

F. Training Procedure

The complete training objective combines audio reconstruction, KL, and attribute supervision terms:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{audio}} + \lambda_{\text{KL}}^c \mathcal{L}_{\text{KL}}^c + \lambda_{\text{KL}}^s \mathcal{L}_{\text{KL}}^s + \lambda_{\text{pitch}} \mathcal{L}_{\text{pitch}} + \lambda_{\text{preset}} \mathcal{L}_{\text{preset}}, \quad (13)$$

where $\mathcal{L}_{\text{audio}} = \lambda_{\text{mel}} \mathcal{L}_{\text{mel}} + \lambda_{\text{adv}} \mathcal{L}_{\text{adv}} + \lambda_{\text{fm}} \mathcal{L}_{\text{fm}}$ combines a multi-scale mel-spectrogram loss with the adversarial and L_1 feature-matching losses against a jointly trained discriminator. The tone masking objective $\mathcal{L}_{\text{mask}}$ is realized through $\mathcal{L}_{\text{audio}}$ in the masked-reconstruction mode. Loss weights and other hyperparameters are given in Section IV-B.

a) *Training modes.*: Each step samples one of three modes, all extracting content from a *content-match* recording $\mathbf{x}_{\text{wet}}^{(c_0, \cdot)}$ and differing in the tone reference and target. *Self-reconstruction* draws the tone from $\mathbf{x}_{\text{wet}}^{(c_1, t_0)}$ and targets $\mathbf{x}_{\text{wet}}^{(c_0, t_0)}$. *Conversion* draws the tone from an external recording $\mathbf{x}_{\text{wet}}^{(c_2, t_2)}$ and targets $\mathbf{x}_{\text{wet}}^{(c_0, t_2)}$, exposing the model to tone references beyond the batch. *Masked reconstruction* masks the tone and targets the dry DI signal $\mathbf{x}_{\text{dry}}$, realizing $\mathcal{L}_{\text{mask}}$.

b) *Two-stage training*: Stage 1 establishes the disentangled factorization under simplified conditions: the tone path is deterministic (no variational head, $\lambda_{\text{KL}}^s = 0$), pvpGD is off, and the data is unaugmented. Mode sampling, $\mathcal{L}_{\text{KL}}^c$, $\mathcal{L}_{\text{pitch}}$, and $\mathcal{L}_{\text{preset}}$ jointly enforce disentanglement and ground the masked state to the dry signal. We omit the variational head here because sampling $\mathbf{s}$ from random initialization destabilizes the disentanglement objectives and hinders convergence. Stage 2 initializes from the stage-1 checkpoint and adds the smoothness components (Section III-E): the variational head produces $(\boldsymbol{\mu}_s, \boldsymbol{\sigma}_s)$ for sampling $\mathbf{s}$, $\mathcal{L}_{\text{KL}}^s$ is warmed up to ease the transition from the deterministic path, pvpGD prevents collapse of the stochastic tone latent, and audio-effects augmentation broadens the tone manifold. Preset classification is disabled ($\lambda_{\text{preset}} = 0$), since augmentation breaks the correspondence between each recording and its preset label.

c) *Inference*: The encoder is deterministic, using $(\boldsymbol{\mu}_c, \boldsymbol{\mu}_s)$ and disabling pvpGD. EGTT combines the content embedding of a source recording with the tone embedding of a reference; EGTR applies the masked forward pass.

IV. EXPERIMENTAL SETUP

A. Datasets

Because no existing electric guitar dataset offers realistic recordings with sufficient tonal diversity, we construct our own using commercial audio plugins from *Neural DSP*.² We take the dry DI signals from the EGDB dataset [1] and render them through two plugins—*Archetype: Cory Wong* X^3 provides the *seen* tones used for training, and *Morgan*

*Amps Suite*⁴ provides the *unseen* tones used for evaluation. Crucially, the two plugins implement different effects and amplifiers, so the evaluation tones are produced by signal chains unseen during training.

Each plugin exposes a complete guitar signal chain spanning all four effect categories of Section II-A: *Archetype: Cory Wong X*, for instance, includes distortion (overdrive, booster, and tube amplifiers with cabinet simulation), modulation (chorus and an envelope filter), equalization (a 9-band graphic EQ), and spatial effects (delay and reverb). We render each plugin’s official factory presets, 218 for *Archetype: Cory Wong X* and 96 for *Morgan Amps Suite*, each a professionally designed tone that exercises a different configuration of this chain. To further increase tonal diversity, we additionally include the EGDB-PG dataset [2], which provides 256 amplifier-only tones.

In total, the EGDB DI signals amount to roughly 2 hours of mono audio. Rendering them through the 256 + 218 training tones (EGDB-PG and *Archetype: Cory Wong X*) yields approximately $2 \times (256 + 218) = 948$ hours for training, and rendering through the 96 *Morgan Amps Suite* presets yields $2 \times 96 = 192$ hours for evaluation. Compared with existing electric guitar datasets, ours offers substantially greater tonal diversity and broader coverage of the signal chain: EGDB [1] provides only 5 presets, EGDB-PG [2] 256, and SynthTab [5] 23, and all three cover amplifiers alone rather than the full pedal-and-amplifier chain. The dataset is split into training, validation, and test sets with an 80/5/15 ratio.

B. Implementation Details

EG-VAE builds on the DAC-VAE backbone. The encoder uses downsampling rates [2, 8, 10, 12] for a hop length of 1920, and the decoder mirrors it with rates [12, 10, 8, 2], operating at a 44.1 kHz sample rate. The content extractor applies its variational bottleneck at dimension 128 before projecting back to the 1024-dimensional latent consumed by the decoder; the tone embedding has dimension 64. The tone extractor is a 4-layer feed-forward Transformer with 4 attention heads, followed by mean pooling and the variational head.

We follow the discriminator and adversarial setup of DAC-VAE: a multi-period waveform discriminator with periods {2, 3, 5, 7, 11} and a multi-scale complex STFT discriminator with FFT sizes {2048, 1024, 512} and band splits at {0, 0.1, 0.25, 0.5, 0.75, 1.0} of the Nyquist frequency. The multi-scale mel loss uses $n_{\text{mels}} \in \{5, 10, 20, 40, 80, 160, 320\}$ with corresponding window lengths {32, 64, ..., 2048}, and the multi-scale STFT loss uses window lengths {2048, 512}.

Both generator and discriminator are optimized with AdamW ($\beta_1 = 0.8$, $\beta_2 = 0.99$, learning rate 1.5×10^{-4}) and an exponential learning-rate decay ($\gamma = 0.9999996$). Models are trained on 1-second segments with a batch size of 14. Stage 1 trains for 150,000 steps; stage 2 fine-tunes from the stage-1 checkpoint for 200,000 steps, with the tone KL weight λ_{KL}^s linearly warmed up over the first 10,000 steps. The three training modes (self-reconstruction, conversion, masked reconstruction) are sampled with probabilities [0.25, 0.45, 0.30]. We

²<https://neuraldsp.com/>

³<https://neuraldsp.com/plugins/archetype-cory-wong>

⁴<https://neuraldsp.com/plugins/morgan-amps-suite>

use loss weights $\lambda_{\text{mel}} = 15$, $\lambda_{\text{fm}} = 2$, $\lambda_{\text{adv}} = 1$, $\lambda_{\text{KL}}^c = \lambda_{\text{KL}}^s = 10^{-4}$, and (in stage 1) $\lambda_{\text{pitch}} = 1$, $\lambda_{\text{preset}} = 5$. The stage-2 audio-effects augmentation (Section III-E) is implemented with the `audiomentations` library⁵ and applied to each preset recording with probability 0.5. All configurations are trained on single NVIDIA RTX PRO 6000 GPU.

C. Baselines

a) *EGTT baselines.*: We compare against two tone-transfer baselines: DeepAFx [14], a differentiable audio-effects style-transfer method, and One-to-many [15], the amplifier-modeling approach. For the strongest DeepAFx configuration, we use a chain spanning all effect categories: gain $\rightarrow$ parametric EQ $\rightarrow$ compressor $\rightarrow$ distortion $\rightarrow$ reverberation. Neither is designed for wet input: One-to-many assumes a dry DI signal, while DeepAFx places no explicit restriction. To probe whether their difficulty stems specifically from wet input, we train each baseline in two configurations. The standard configuration takes the wet recording directly as input. The *EGTR-enhanced* configuration (denoted *w/ EGTR*) is trained on dry input and, at inference, sources from a dry signal recovered by EG-VAE’s tone removal. Since EG-VAE attains the best removal performance among all methods (Table II), this gives the baselines the strongest available dry estimate, relieving them of the wet-input mismatch (Section II-B) for the most favorable comparison.

b) *EGTR baselines.*: We compare against four tone-removal baselines spanning the two lines discussed in Section II-C. HDemucs [44], DCUNet [45], and DPTNet [46] are source-separation and enhancement architectures adapted to recover the dry DI signal, and Distortion Recovery [17] is a specialized model for guitar effect removal. We follow the baseline selection of prior removal work [17], [18].

c) *Training setup.*: All baselines are retrained on the same data as EG-VAE, following the architecture and training setup of each original work, with the optimization objective standardized to the multi-scale mel-spectrogram loss used in DAC-VAE for a consistent comparison across methods. For Distortion Recovery [17], we additionally replace the original HiFi-GAN vocoder with the more recent BigVGAN [47] for higher-fidelity synthesis.

D. Evaluation Protocol

We evaluate EG-VAE along five axes: tone transfer (EGTT, Table I), tone removal (EGTR, Table II), tone-space smoothness (Table V), an ablation study over its components (Tables III and IV), and a qualitative analysis of the learned representation. Unless otherwise noted, experiments report results on both *seen* and *unseen* tones, with all objective metrics averaged over 2,000 ten-second test examples.

a) *Objective metrics.*: Following [34], we report two spectral distances between the model output and the task target: the *Mel distance*, the L_1 distance between log-mel spectrograms, and the *STFT distance*, the L_1 distance between log-magnitude spectrograms, which better captures fidelity at

higher frequencies. The target is the reference-tone recording for EGTT and the dry DI signal for EGTR; lower values indicate closer agreement.

b) *Subjective metrics.*: As spectral distances do not fully reflect perceptual quality, we also conduct a listening test with 11 participants, rating each example on a 1–5 scale. For both tasks, listeners rate *audio quality* (AQ), the overall fidelity and naturalness of the output, independent of the task objective. Each task adds a task-specific criterion: *tone similarity* for EGTT, measuring how closely the output’s tone matches the reference tone, and *dryness* for EGTR, measuring how closely the output resembles a clean DI signal, i.e., how completely the signal-chain processing has been removed. We additionally include the ground-truth target as an upper-bound anchor (denoted *Oracle* in Tables I and II), providing a reference for the highest rating listeners assign under each criterion.

c) *Tone-space smoothness.*: To assess whether the tone space is smooth, we adopt the perceptual path length (PPL) [48], which measures the perceptual change in the decoded output under small displacements along an interpolation path in the embedding space. Concretely, we interpolate between the tone embeddings of two reference recordings while holding the content fixed; at 32 random positions t along the path, we measure the distance between outputs decoded at t and $t + \epsilon$. We evaluate over 500 such interpolation pairs, yielding $500 \times 32 = 16,000$ samples. Following [48], we normalize each distance by ϵ^2 (with $\epsilon = 10^{-4}$); as the perceptual distance we use the L_1 distance between normalized log-mel spectrograms. A lower PPL indicates that small steps in tone space produce small output changes, i.e., a smoother space. We report PPL on unseen tones, the setting smoothness is designed to support.

d) *Ablation study.*: We ablate EG-VAE in two groups, reflecting the two stages of training. The first (Table III) removes individual disentanglement mechanisms active in stage 1: MIDI (pitch) supervision, tone (preset) classification, content–tone perturbation, and tone masking, and evaluates each variant across three modes: reconstruction, tone transfer, and tone removal. As these mechanisms govern the factorization itself rather than generalization, we evaluate this group on seen tones only. Removing tone masking (*w/o Tone Masking*) disables EGTR entirely, so its removal column is left empty. The second group (Table IV) ablates the stage-2 components against the full model: variational sampling, audio-effects augmentation, and pvpGD. pvpGD appears here rather than in the first group because, as a mechanism guarding against posterior collapse, it is only operative once the variational tone path is introduced in stage 2 (Section III-C).

e) *Representation visualization.*: To assess the factorization qualitatively, we visualize tone embeddings on seen and unseen reference tones with t-SNE [49]. A well-disentangled representation should yield tone embeddings that cluster by preset, capturing tone identity.

V. RESULTS AND DISCUSSION

A. Tone Transfer (EGTT)

Table I reports tone transfer results. EG-VAE achieves the lowest spectral distances by a clear margin, reaching 0.86 Mel

⁵<https://github.com/iver56/audiomentations>

TABLE I

TONE TRANSFER RESULTS ON SEEN AND UNSEEN TONES. MEL AND STFT ARE OBJECTIVE SPECTRAL DISTANCES, WHILE AQ (AUDIO QUALITY) AND TONE SIM. ARE SUBJECTIVE RATINGS FROM THE LISTENING TEST.

Model	Seen Tone / Archetype: Cory Wong X				Unseen Tone / Morgan Amps Suite			
	Mel ↓	STFT ↓	AQ ↑	Tone Sim. ↑	Mel ↓	STFT ↓	AQ ↑	Tone Sim. ↑
Oracle	-	-	4.26 ± 0.71	4.48 ± 0.64	-	-	3.85 ± 1.06	4.48 ± 0.70
DeepAFx [14]	2.07 ± 0.74	3.0 ± 0.84	2.85 ± 1.23	1.44 ± 0.58	1.63 ± 0.49	2.58 ± 0.57	2.19 ± 0.83	1.81 ± 0.83
w/ EGTR	2.04 ± 0.64	2.98 ± 0.73	3.37 ± 1.08	2.30 ± 1.23	1.75 ± 0.43	2.72 ± 0.49	3.15 ± 0.91	2.74 ± 1.20
One-to-many [15]	1.93 ± 0.62	2.85 ± 0.83	3.67 ± 1.11	2.30 ± 0.82	1.69 ± 0.44	2.7 ± 0.64	3.67 ± 0.78	2.63 ± 0.93
w/ EGTR	1.53 ± 0.42	2.55 ± 0.83	3.11 ± 0.89	2.32 ± 0.99	1.61 ± 0.35	2.61 ± 0.64	2.56 ± 0.97	2.07 ± 0.78
EG-VAE	0.86 ± 0.17	1.63 ± 0.15	4.15 ± 0.66	4.30 ± 0.61	1.15 ± 0.24	2.00 ± 0.28	3.85 ± 0.82	3.70 ± 0.72

TABLE II

TONE REMOVAL RESULTS ON SEEN AND UNSEEN TONES. MEL AND STFT ARE OBJECTIVE SPECTRAL DISTANCES, WHILE AQ (AUDIO QUALITY) AND DRYNESS ARE SUBJECTIVE RATINGS FROM THE LISTENING TEST.

Model	Seen Tone / Archetype: Cory Wong X				Unseen Tone / Morgan Amps Suite			
	Mel ↓	STFT ↓	AQ ↑	Dryness ↑	Mel ↓	STFT ↓	AQ ↑	Dryness ↑
Oracle	-	-	4.78 ± 0.42	4.67 ± 0.48	-	-	4.44 ± 0.80	4.44 ± 0.58
HDemucs [44]	1.36 ± 0.39	2.48 ± 0.89	2.96 ± 1.29	2.70 ± 1.10	1.63 ± 0.39	2.63 ± 0.88	3.00 ± 1.04	2.37 ± 1.21
DCUNet [45]	1.54 ± 0.47	2.77 ± 0.91	3.37 ± 1.04	2.93 ± 0.92	1.57 ± 0.43	2.71 ± 0.85	3.19 ± 1.11	2.78 ± 1.09
DPTNet [46]	1.19 ± 0.40	2.01 ± 0.61	2.70 ± 1.03	2.22 ± 0.89	1.27 ± 0.37	2.06 ± 0.63	2.67 ± 1.30	2.41 ± 1.31
Distortion Recovery [17]	1.21 ± 0.34	1.85 ± 0.61	1.52 ± 0.58	2.48 ± 1.19	1.22 ± 0.34	1.86 ± 0.62	2.04 ± 0.98	2.22 ± 1.15
EG-VAE	1.10 ± 0.24	1.79 ± 0.43	3.48 ± 0.75	3.48 ± 0.75	1.19 ± 0.28	1.87 ± 0.50	3.70 ± 0.87	3.48 ± 0.70

and 1.63 STFT on seen tones versus 1.53 and 2.55 for the strongest baseline (One-to-many w/ EGTR), and maintaining its lead on unseen tones (1.15 Mel, 2.00 STFT) despite a slight drop relative to seen tones.

The EGTR-enhanced baselines confirm our analysis of the wet-input setting (Section II-B): supplying a tone-removed input improves both methods on seen tones (One-to-many $1.93 \rightarrow 1.53$ Mel, DeepAFx $2.07 \rightarrow 2.04$), yet both remain well behind EG-VAE. We attribute this gap to the two-stage nature of the baselines, each difficult in its own right. DeepAFx relies on a differentiable chain of fixed type and order, which cannot model chains deviating from its structure, while One-to-many’s more flexible gated convolutional backbone still does not capture the full diversity of guitar tones. A two-stage pipeline thus compounds removal and modeling errors, whereas EG-VAE transfers on a single representation, avoiding this accumulation.

Notably, the weaker baselines score lower on unseen than on seen tones, the opposite of EG-VAE. This reflects their weak fit in both settings rather than better generalization: having fit the seen tones poorly, they have no seen-tone advantage to lose, whereas EG-VAE’s seen-best ordering is the expected signature of a model that fits the trained tones and degrades gracefully on held-out ones.

Subjective ratings show similar trends with objective results. EG-VAE achieves the highest AQ and tone similarity on both seen (4.15, 4.30) and unseen (3.85, 3.70) tones, approaching the Oracle reference (4.26/4.48 seen, 3.85/4.48 unseen) and demonstrating strong generalization. On AQ, all baselines remain reasonably close to EG-VAE—unsurprisingly, since effects-transfer methods fundamentally preserve audio fidelity.

The main gap lies in tone similarity, where the baselines cluster at low scores (1.81–2.74) with no consistent preference among them, and EG-VAE nearly doubles the next-best score on both settings. The gap widens further on unseen tones: even One-to-many w/ EGTR, the objectively strongest baseline on seen tones, drops to 2.07, reflecting the difficulty for amplifier-targeted methods of applying effects beyond their training distribution. By contrast, EG-VAE’s smaller drop (4.30 $\rightarrow$ 3.70) reflects the smooth tone space shaped in stage 2 (Section V-D).

B. Tone Removal (EGTR)

Table II reports tone removal results. On seen tones, EG-VAE attains the best spectral distances (1.10 Mel, 1.79 STFT), outperforming all removal baselines. On unseen tones, it remains strongest in Mel distance (1.19) and is comparable to Distortion Recovery in STFT (1.87 versus 1.86). This shows the two tasks can be unified without sacrificing either.

Figure 4 illustrates this on a distortion-heavy example. The difficulty lies in the harmonic region marked by the red boxes: nonlinear distortion introduces dense high-order harmonics absent from the dry DI. In this region, the baselines leave large residuals (right column), either over-suppressing or failing to remove the added harmonics, whereas EG-VAE’s residual is visibly smaller. This reflects the ill-posedness of inverting nonlinear distortion (Section II-C): the added harmonics are hard to separate from the underlying signal, and EG-VAE’s content representation, trained to reconstruct the dry DI under tone masking, captures it more accurately than the enhancement-based baselines.

TABLE III
ABLATION STUDY RESULTS ACROSS THREE EVALUATION MODES. VALUES ARE REPORTED AS MEAN $\pm$ STD.

Model	Reconstruction		Tone Transfer		Tone Removal	
	Mel $\downarrow$	STFT $\downarrow$	Mel $\downarrow$	STFT $\downarrow$	Mel $\downarrow$	STFT $\downarrow$
EG-VAE	0.80 $\pm$ 0.12	1.58 $\pm$ 0.13	0.86 $\pm$ 0.16	1.64 $\pm$ 0.16	1.13 $\pm$ 0.27	1.84 $\pm$ 0.49
<i>Ablation Studies</i>						
w/o MIDI	1.02 $\pm$ 0.18	1.81 $\pm$ 0.16	1.10 $\pm$ 0.25	1.89 $\pm$ 0.21	1.56 $\pm$ 0.30	2.37 $\pm$ 0.48
w/o Tone Classification	0.85 $\pm$ 0.15	1.64 $\pm$ 0.15	0.95 $\pm$ 0.23	1.73 $\pm$ 0.21	1.21 $\pm$ 0.28	1.97 $\pm$ 0.51
w/o Perturbation	1.48 $\pm$ 0.39	2.57 $\pm$ 0.44	1.67 $\pm$ 0.51	2.74 $\pm$ 0.56	2.12 $\pm$ 0.56	3.15 $\pm$ 1.17
w/o Tone Masking	0.96 $\pm$ 0.18	1.76 $\pm$ 0.15	1.02 $\pm$ 0.23	1.83 $\pm$ 0.20	-	-

TABLE IV
ABLATION STUDY RESULTS ACROSS THREE EVALUATION MODES.

Model	Reconstruction		Tone Transfer		Tone Removal	
	Mel $\downarrow$	STFT $\downarrow$	Mel $\downarrow$	STFT $\downarrow$	Mel $\downarrow$	STFT $\downarrow$
EG-VAE (two stage)	1.02 $\pm$ 0.15	1.88 $\pm$ 0.24	1.14 $\pm$ 0.21	2.00 $\pm$ 0.28	1.18 $\pm$ 0.27	1.86 $\pm$ 0.49
<i>Ablation Studies</i>						
w/o Variational Sampling	1.10 $\pm$ 0.17	1.96 $\pm$ 0.27	1.25 $\pm$ 0.23	2.10 $\pm$ 0.31	1.24 $\pm$ 0.30	1.94 $\pm$ 0.54
w/o Fx-Aug	1.10 $\pm$ 0.16	1.99 $\pm$ 0.27	1.23 $\pm$ 0.23	2.10 $\pm$ 0.31	1.25 $\pm$ 0.31	1.95 $\pm$ 0.58
w/o PvpGD	1.01 $\pm$ 0.16	1.89 $\pm$ 0.28	1.15 $\pm$ 0.23	2.02 $\pm$ 0.32	1.20 $\pm$ 0.30	1.88 $\pm$ 0.54

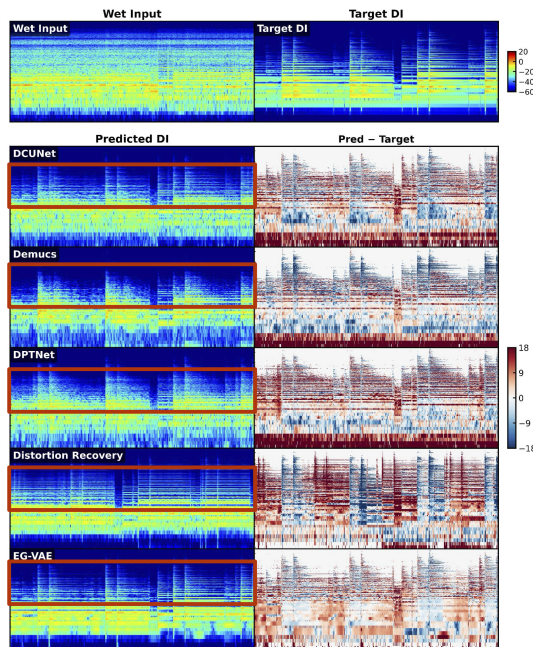


Fig. 4. Tone removal on a representative distortion-heavy tone. Top: wet input and target DI. For each method: predicted DI (left, log-magnitude dB) and the residual between prediction and target.

The subjective results reveal a notable divergence from the objective ranking. While Distortion Recovery achieves competitive spectral scores (Mel 1.21, STFT 1.85 on seen), its perceived quality is the lowest among all methods (1.52 AQ on seen, 2.04 on unseen), indicating that minimizing spectral distance alone does not yield a perceptually clean DI signal. EG-VAE, by contrast, leads on both AQ (3.48 seen, 3.70 unseen) and dryness (3.48 both settings), approaching

the Oracle (4.67, 4.44 dryness) and remaining stable across seen and unseen tones—suggesting that joint training under tone masking produces outputs listeners consistently judge as both faithful and dry.

C. Ablation Study

a) *Disentanglement mechanisms*: Table III ablates the disentanglement mechanisms across the reconstruction, tone transfer, and tone-removal modes. Every mechanism contributes, removing any one degrades performance, confirming that each plays a distinct role in the factorization. Content-tone perturbation is the most apparent: removing it degrades all three modes most severely (reconstruction Mel 0.80 $\rightarrow$ 1.48, tone removal 1.13 $\rightarrow$ 2.12), as this strategy is what forces each embedding to retain only its intended factor; without it, content and tone are not factorized into the desired embedding. Removing MIDI supervision also harms all modes substantially (reconstruction Mel 0.80 $\rightarrow$ 1.02, tone removal 1.13 $\rightarrow$ 1.56). Distortion obscures the harmonic structure of the played notes in the spectrogram, making the content hard to recover; supervising $\mathbf{z}_{1:T'}$ with per-frame pitch counteracts this by explicitly preserving the playing content. Removing preset classification has a smaller effect on reconstruction and tone transfer but still degrades tone removal (1.13 $\rightarrow$ 1.21 Mel), consistent with its role in shaping the tone embedding rather than the content path.

Removing tone masking disables tone removal entirely (shown as “-”), since masking is the mechanism by which removal is performed. It also degrades reconstruction and tone transfer (0.80 $\rightarrow$ 0.96 Mel reconstruction), showing that masking is not merely a removal procedure but also strengthens the content-tone factorization.

b) *Stage-2 components*: Table IV ablates the stage-2 components. Removing variational sampling degrades most

TABLE V
TONE-SPACE SMOOTHNESS ON UNSEEN TONES. LOWER IS SMOOTHER.

Model	Smoothness ↓
EG-VAE	0.90
<i>Ablation Studies</i>	
w/o Variational Sampling	2.00
w/o Fx-Aug	0.91
w/o PvpGD	0.96

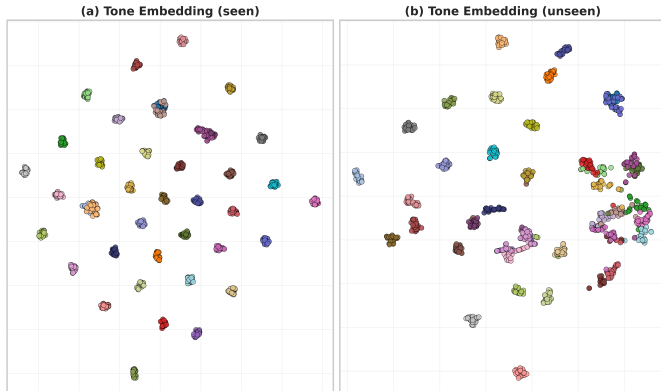


Fig. 5. t-SNE visualization of the tone embedding s , colored by preset.

modes, most notably tone transfer (Mel 1.14 $\rightarrow$ 1.25, STFT 2.00 $\rightarrow$ 2.10) and tone-removal STFT (1.86 $\rightarrow$ 1.94). The KL regularization $\mathcal{L}_{\text{KL}}^s$ organizes the tone embeddings into a continuous, Gaussian-structured manifold, so that tones near the training distribution, including unseen ones, decode coherently. A deterministic embedding lacks this structure and generalizes worse to held-out chains. Removing audio-effects augmentation similarly degrades tone transfer and removal (e.g., removal Mel 1.18 $\rightarrow$ 1.25), consistent with its role in broadening the tone space toward unseen configurations. Removing pvpGD produces a small but consistent degradation in tone transfer, removal, and smoothness (Tables IV, V) at a negligible reconstruction cost: by coupling the two latents to prevent collapse of s , it keeps the tone embedding informative, trading marginal reconstruction fidelity for a more usable tone representation.

D. Tone-Space Smoothness

Table V reports the perceptual path length (PPL) on unseen tones, measuring how smoothly the decoded output varies along interpolation paths in the tone space; lower values indicate a smoother space. EG-VAE achieves the smoothest space (0.90), and removing any stage-2 component increases PPL. The effect is by far largest for variational sampling (0.90 $\rightarrow$ 2.00), confirming it as the primary driver of smoothness: without the stochastic, KL-regularized tone path, interpolating between two tones produces abrupt, less coherent changes in the output. This matches the design rationale of Section III-E, where sampling s under KL regularization is the mechanism that makes the decoder vary continuously with the tone embedding.

Removing audio-effects augmentation or pvpGD increases PPL only marginally (0.91 and 0.96 versus 0.90). Both contribute to smoothness, but far less than variational sampling, which is consistent with their distinct roles: augmentation broadens the region covered by the tone space, and pvpGD keeps s informative by preventing collapse, whereas variational sampling shapes the continuous structure on which these two operate.

E. Representation Visualization

To qualitatively examine the tone representation, we visualize the tone embeddings s with t-SNE in Fig. 5, coloring each point by its preset label. For seen tones (Fig. 5a), the embeddings form clear, well-separated clusters: recordings sharing a signal chain map to nearby points while different presets occupy distinct regions, indicating that s captures tone identity. For unseen tones (Fig. 5b), the clusters remain identifiable but less separated, consistent with the seen–unseen performance gap in transfer and removal.

VI. CONCLUSION

We presented EG-VAE, a unified framework for electric guitar tone transfer (EGTT) and tone removal (EGTR). Rather than treating the two tasks separately, EG-VAE learns a single disentangled representation of wet recordings, factorizing them into a content embedding and a tone embedding and realizing both tasks. Transfer recombines content with a reference tone, while removal is realized by a novel tone masking objective that plays a dual role: it strengthens content–tone disentanglement during training and defines the removal procedure at inference, using one operation identical in both. A second training stage shapes a smooth tone space, improving transfer quality on tones unseen during training.

Several directions remain for future work. The content embedding could be factorized further into the played notes (e.g., MIDI) and the playing performance, enabling independent control over what is played and how. A harder challenge is delay: a delayed repetition is physically an effect yet acoustically resembles performed content, giving the model no consistent cue for assigning it to tone or content, an open problem for tone disentanglement. Finally, extending the framework to other instruments would broaden its applicability.

REFERENCES

- [1] Y.-H. Chen, W.-Y. Hsiao, T.-K. Hsieh, J.-S. R. Jang, and Y.-H. Yang, “Towards automatic transcription of polyphonic electric guitar music: A new dataset and a multi-loss transformer model,” in *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, 2022.
- [2] Y.-H. Chen, Y.-C. Cheng, Y.-T. Yeh, J.-T. Wu, J.-S. R. Jang, and Y.-H. Yang, “Towards generalizability to tone and content variations in the transcription of amplifier rendered electric guitar audio,” *arXiv preprint arXiv:2504.07406*, 2025.
- [3] X. Riley, D. Edwards, and S. Dixon, “High resolution guitar transcription via domain adaptation,” in *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2024.
- [4] M. Stein, J. Abeßer, C. Dittmar, and G. Schuller, “Automatic detection of audio effects in guitar and bass recordings,” in *Audio Engineering Society Convention 128*, 2010.

- [5] Y. Zang, Y. Zhong, F. Cwitkowitz, and Z. Duan, "Synthtab: Leveraging synthesized data for guitar tablature transcription," in *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2024.
- [6] J. Loth, P. Sarmiento, M. Sandler, and M. Barthet, "Guitarflow: Realistic electric guitar synthesis from tablatures via flow matching and style transfer," *arXiv preprint arXiv:2510.21872*, 2025.
- [7] F. Eichas, S. Möller, and U. Zölzer, "Block-oriented gray box modeling of guitar amplifiers," in *Proc. International Conference on Digital Audio Effects (DAFx)*, 2017.
- [8] A. Wright, E.-P. Damskägg, and V. Välimäki, "Real-time black-box modelling with recurrent neural networks," in *Proc. International Conference on Digital Audio Effects (DAFx)*, 2019.
- [9] A. Wright and V. Valimäki, "Neural modeling of phaser and flanging effects," *Journal of the Audio Engineering Society*, 2021.
- [10] Y.-T. Yeh, Y.-H. Chen, Y.-C. Cheng, J.-T. Wu, J.-J. Fu, Y.-F. Yeh, and Y.-H. Yang, "Ddsp guitar amp: Interpretable guitar amplifier modeling," in *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2025.
- [11] C.-Y. Yu, M. A. Martínez-Ramírez, J. Koo, W.-H. Liao, Y. Mitsufuji, and G. Fazekas, "Improving inference-time optimisation for vocal effects style transfer with a gaussian prior," in *2025 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2025.
- [12] C. J. Steinmetz, S. Singh, M. Comunità, I. Ibnyahya, S. Yuan, E. Benetos, and J. D. Reiss, "St-ito: Controlling audio effects for style transfer with inference-time optimization," in *Proc. International Society for Music Information Retrieval (ISMIR)*, 2024.
- [13] J. Koo, M. A. Martínez-Ramírez, W.-H. Liao, G. Fabbro, M. Mancusi, and Y. Mitsufuji, "Ito-master: Inference-time optimization for audio effects modeling of music mastering processors," in *Proc. International Society for Music Information Retrieval (ISMIR)*, 2025.
- [14] C. J. Steinmetz, N. J. Bryan, and J. D. Reiss, "Style transfer of audio effects with differentiable signal processing," *Journal of the Audio Engineering Society*, 2022.
- [15] Y.-H. Chen, Y.-T. Yeh, Y.-C. Cheng, J.-T. Wu, Y.-H. Ho, J.-S. R. Jang, and Y.-H. Yang, "Towards zero-shot amplifier modeling: One-to-many amplifier modeling via tone embedding control," in *Proc. International Society for Music Information Retrieval (ISMIR)*, 2024.
- [16] J. Imort, G. Fabbro, M. A. M. Ramírez, S. Uhlich, Y. Koyama, and Y. Mitsufuji, "Distortion audio effects: Learning how to recover the clean signal," in *Proc. International Society for Music Information Retrieval (ISMIR)*, 2022.
- [17] Y.-S. Lee, Y.-P. Peng, J.-T. Wu, M. Cheng, L. Su, and Y.-H. Yang, "Distortion recovery: A two-stage method for guitar effect removal," in *Proc. International Conference on Digital Audio Effects (DAFx)*, 2024.
- [18] M. Rice, C. J. Steinmetz, G. Fazekas, and J. D. Reiss, "General purpose audio effect removal," in *2023 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2023.
- [19] Y.-J. Luo and S. Dixon, "Posterior variance-parameterised gaussian dropout: Improving disentangled sequential autoencoders for zero-shot voice conversion," in *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2024.
- [20] D. T. Yeh, J. S. Abel, and J. O. Smith, "Automated physical modeling of nonlinear audio circuits for real-time audio effects," *IEEE Trans. Audio, Speech, Lang. Process.*, 2010.
- [21] J. Pakarinen and D. T. Yeh, "A review of digital techniques for modeling vacuum-tube guitar amplifiers," *Computer Music Journal*, 2009.
- [22] J. Engel, L. Hantrakul, C. Gu, and A. Roberts, "Ddsp: Differentiable digital signal processing," in *Proc. International Conference on Learning Representations (ICLR)*, 2020.
- [23] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," in *Proc. NeurIPS*, 2020.
- [24] K. Qian, Y. Zhang, S. Chang, X. Yang, and M. Hasegawa-Johnson, "Autovc: Zero-shot voice style transfer with only autoencoder loss," in *International Conference on Machine Learning*, 2019.
- [25] Z. Ju, Y. Wang, K. Shen, X. Tan, D. Xin, D. Yang, Y. Liu, Y. Leng, K. Song, S. Tang *et al.*, "Natalspeech 3: Zero-shot speech synthesis with factorized codec and diffusion models," in *Proc. ICML*, 2024.
- [26] X. Zhang, X. Zhang, K. Peng, Z. Tang, V. Manohar, Y. Liu, J. Hwang, D. Li, Y. Wang, J. Chan *et al.*, "Vevo: Controllable zero-shot voice imitation with self-supervised disentanglement," in *Proc. International Conference on Learning Representations (ICLR)*, 2025.
- [27] X. Zhang, J. Zhang, Y. Wang, C. Wang, Y. Chen, D. Jia, Z. Chen, and Z. Wu, "Vevo2: A unified and controllable framework for speech and singing voice generation," *arXiv preprint arXiv:2508.16332*, 2025.
- [28] C. H. Lee, J. Nistal, S. Lattner, M. Pasini, and G. Fazekas, "Diffusion timbre transfer via mutual information guided inpainting," in *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2026.
- [29] T. Baoueb, X. Bie, H. Janati, and G. Richard, "Wavetransfer: A flexible end-to-end multi-instrument timbre transfer with diffusion," in *2024 IEEE 34th International Workshop on Machine Learning for Signal Processing (MLSP)*, 2024.
- [30] Y. Li and S. Mandt, "Disentangled sequential autoencoder," in *Proc. ICML*, 2018.
- [31] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.
- [32] I. Higgins, L. Matthey, A. Pal, C. P. Burgess, X. Glorot, M. M. Botvinick, S. Mohamed, and A. Lerchner, "beta-vae: Learning basic visual concepts with a constrained variational framework," in *Proc. International Conference on Learning Representations (ICLR)*, 2017.
- [33] A. Polyak, A. Zohar, A. Brown, A. Tjandra, A. Sinha, A. Lee, A. Vyas, B. Shi, C.-Y. Ma, C.-Y. Chuang *et al.*, "Movie gen: A cast of media foundation models," *arXiv preprint arXiv:2410.13720*, 2024.
- [34] R. Kumar, P. Seetharaman, A. Luebs, I. Kumar, and K. Kumar, "High-fidelity audio compression with improved rvqgan," in *Proc. NeurIPS*, 2023.
- [35] M. Chen, X. Tan, B. Li, Y. Liu, T. Qin, S. Zhao, and T.-Y. Liu, "Adaspeech: Adaptive text to speech for custom voice," in *Proc. International Conference on Learning Representations (ICLR)*, 2021.
- [36] Y.-J. Luo, K. Agres, and D. Herremans, "Learning disentangled representations of timbre and pitch for musical instrument sounds using gaussian mixture variational autoencoders," in *Proc. International Society for Music Information Retrieval (ISMIR)*, 2019.
- [37] L. Ziyin, T. Hartwig, and M. Ueda, "Neural networks fail to learn periodic functions and how to fix it," *Advances in Neural Information Processing Systems*, 2020.
- [38] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.
- [39] H.-S. Choi, J. Lee, W. Kim, J. Lee, H. Heo, and K. Lee, "Neural analysis and synthesis: Reconstructing speech from self-supervised representations," in *Proc. NeurIPS*, 2021.
- [40] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1501–1510.
- [41] J.-Y. Liu, T.-C. Hsu, Y.-T. Yeh, L. Su, and Y.-H. Yang, "Synthcloner: Synthesizer-style audio transfer via factorized codec with adsr envelope control," in *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2026.
- [42] D. Stanton, M. Shannon, S. Mariooryad, R. Skerry-Ryan, E. Battenberg, T. Bagby, and D. Kao, "Speaker generation," in *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2022.
- [43] W.-N. Hsu, Y. Zhang, R. J. Weiss, H. Zen, Y. Wu, Y. Wang, Y. Cao, Y. Jia, Z. Chen, J. Shen *et al.*, "Hierarchical generative modeling for controllable speech synthesis," in *Proc. International Conference on Learning Representations (ICLR)*, 2018.
- [44] A. Défossez, "Hybrid spectrogram and waveform source separation," *arXiv preprint arXiv:2111.03600*, 2021.
- [45] H.-S. Choi, J.-H. Kim, J. Huh, A. Kim, J.-W. Ha, and K. Lee, "Phase-aware speech enhancement with deep complex u-net," in *Proc. International Conference on Learning Representations (ICLR)*, 2018.
- [46] J. Chen, Q. Mao, and D. Liu, "Dual-path transformer network: Direct context-aware modeling for end-to-end monaural speech separation," in *Proc. Interspeech*, 2020.
- [47] S.-g. Lee, W. Ping, B. Ginsburg, B. Catanzaro, and S. Yoon, "Bigvgan: A universal neural vocoder with large-scale training," in *Proc. International Conference on Learning Representations (ICLR)*, 2022.
- [48] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019.
- [49] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, 2008.