CLEAN2FX: LABEL-CONDITIONED MODELING FOR CLEAN-TO-EFFECT GUITAR AUDIO TRANSFORMATIONS

Oliverio Bombicci Pontelli
School of Electronic Engineering & Computer Science
Queen Mary University of London
London, United Kingdom
oliveriobp@gmail.com

Iran R. Roman
School of Electronic Engineering & Computer Science
Queen Mary University of London
London, United Kingdom
i.roman@qmul.ac.uk

ABSTRACT

We present Clean2FX, a study and demo of label-conditioned clean-to-effect transformation for electric guitar audio. Given a clean guitar input and a target effect label, the task is to synthesize the corresponding effected signal while preserving the musical content. Training and evaluation pairs are constructed from EGFxSet real, single-tone recordings by assembling matched clean/effected chords, melodies, and mixed timelines. This allows for controlled comparison across effects. We evaluate four neural approaches under a common spectrogram-based transformation setting: two variational autoencoders and two U-Net models that differ in whether they operate on linear or log-magnitude representations. Performance is measured using linear-magnitude spectrogram MSE and Fréchet Audio Distance. The U-Net models outperform the variational autoencoder variants. Per-effect results show that distortion effects are most readily improved, whereas delay and reverb effects exhibit weaker FAD gains despite substantial spectral-error reductions. A conditioning-sensitivity diagnostic provides evidence that the best model responds to target labels rather than collapsing to a single transformation. Our demo website compares two models applied on real-world guitar performances outside training and validation data, providing audio and spectrogram examples of the practical clean-to-effect behavior.

1. INTRODUCTION

Effects are central to the sound of the electric guitar: they reshape timbre, dynamics, sustain, and apparent space, and have become part of both the instrument's musical vocabulary and the broader history of audio effects processing [1, 2, 3, 4]. From a signal-processing perspective, effects can be understood as transformations of a clean instrumental source [5]. Recent neural approaches to audio effects have shown that input–output behavior can be learned directly from paired recordings, including black-box modeling of audio effects [6], real-time guitar-amplifier emulation [7], and controllable audio-effect transformation [8]. Much of this work, however, is organized around a single device, a single effect class, or a parametric processor. Clean-to-effect guitar performance transformation across several effect families poses a problem: preservation of pitch, rhythm, and playing content while applying a requested transformation whose acoustic footprint may range from broadband distortion to modulation, delay, or reverb.

This paper presents Clean2FX as both a study of label-conditioned clean-to-effect guitar audio transformation and a web demo. We construct paired clean/effected examples from EGFxSet, a dataset of electric-guitar tones processed through real hardware effects [9, 10], by assembling matched chords, melodies, and mixed timelines. This gives a controlled setting in which the musical content is held fixed and the target effect label defines the transformation. We compare four neural approaches: two variational autoencoders and two conditional encoder–decoder U-Nets.

The demo website complements this controlled evaluation by applying two trained systems to real-world guitar performances [11] outside the EGFxSet-derived material. The demo therefore illustrates the practical behavior of the models beyond synthetic data.[1] The code implementation is released too.[2]

2. METHODS

We build our training data using EGFxSet [9], which contains isolated electric guitar tones recorded through ten real hardware effects alongside a clean reference signal. We want to train models on musical sequences, so we create synthetic performances programmatically, using these recorded tone sounds. Our resulting custom dataset assembles chords, melodies, and mixed event timelines, applying the same event sequence to both the clean and effected versions of each tone. This pairing holds all musical content constant across both signals, leaving the effect as the sole axis of variation in each training example. Each pair is normalized by the larger of its two signal maxima to preserve the energy ratio between the clean and effected signals. The probability of sampling near-silent compositions is set to 0.05.

Audio is represented as magnitude STFT spectrograms with $n_{\text{fft}} = 512$ and hop $h = 128$ at 16 kHz; each 4.9 s clip yields a 257×613 spectrogram. Two input representations are compared: the linear shared-max normalized magnitude x, and a log-compressed variant $y = \ln(1 + 200x)$, inverted as $x = (e^y - 1)/200$ before any metric is computed. Since only magnitude is retained, the phase of the output spectrogram is unavailable; at inference, it is estimated using Griffin-Lim [12].

We assess four architectures, each conditioned on a learned effect embedding. Figure 1 summarizes and illustrates them. The VAE encodes the input with fully connected layers into a 64-dimensional latent and decodes symmetrically, trained with the standard VAE objective (MSE reconstruction plus a KL term). The ConvVAE replaces the dense layers with strided convolutions using BatchNorm and ReLU, a 512-dimensional deterministic latent, and an L1 loss on $\log(1 + x)$ magnitudes. We compare these

[1] guitar-clean2fx-demo.netlify.app
[2] github.com/oliveriobp/fyp-clean2fx

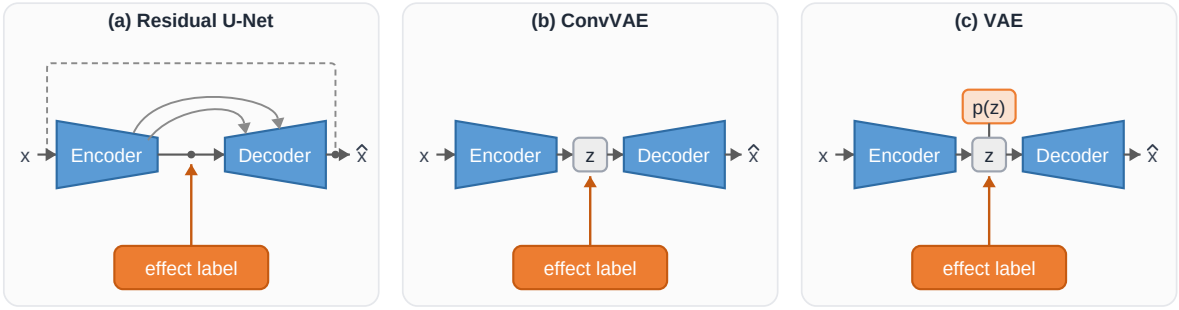

Figure 1: The three model families compared in this work. The U-Net and ConvVAE have a convolutional encoder–decoder structure, conditioned on the target effect label. (a) Residual U-Net: skip connections link encoder and decoder, FiLM modulates the decoder levels and the skip connections, and a global residual path adds the clean input. (b) ConvVAE: a deterministic latent z and no skip connections. (c) VAE: uses dense layers, and the latent z is sampled from $p(z)$. x and $\hat{x}$ are the input and output magnitude spectrograms, respectively.

Table 1: Evaluation of models conditioned to transform clean electric guitar recordings into target effects from EGFxSet [9] (a dataset of real hardware-processed guitar tones). The unprocessed input baseline measures the error when comparing against the clean recording unchanged (i.e., applying no effect) and defines the minimum bar a useful model must exceed. The two variants differ only in whether the loss is applied to linear ($\mathcal{L}_{\text{lin}}$) or log-magnitude ($\mathcal{L}_{\text{log}}$) spectra. MSE: mean squared error on linear shared-max magnitude spectrograms ($\times 10^{-4}$; ↓ better). FAD: Fréchet Audio Distance, a perceptual quality metric (↓ better). Δ: mean per-effect improvement over the unprocessed input baseline (↑ better); "no imp" indicates the model is, on average across effects, worse than applying no effect.

	↓ MSE$_{(\times 10^{-4})}$		↓ FAD	
Model	Value	Δ	Value	Δ
Baseline				
Unprocessed input	5.16	—	6.82	—
Learned models				
VAE	11.44	no imp	17.53	no imp
ConvVAE	4.09	no imp	7.38	no imp
U-Net ($\mathcal{L}_{\text{lin}}$)	1.08	+69.4%	3.98	+23.0%
U-Net ($\mathcal{L}_{\text{log}}$)	**1.07**	**+70.6%**	**3.45**	**+31.8%**

VAEs against a five-level residual U-Net [13, 14]. The encoder comprises four DoubleConv blocks (channels 32, 64, 128, 256); each two 3×3 convolutions with Group Normalization (eight groups [15]) and LeakyReLU (slope 0.01), separated by 2×2 max-pooling; the bottleneck adds a fifth block at 512 channels; the decoder mirrors the encoder with nearest-neighbor upsampling and skip connections. The network predicts a residual summed with the input [16], passed through softplus ($\beta = 5$ for linear inputs, $\beta = 1$ for log) to produce non-negative magnitudes. Two U-Net variants are evaluated, differing only in whether the loss operates on linear ($\mathcal{L}_{\text{lin}}$) or log-magnitude ($\mathcal{L}_{\text{log}}$) spectrograms. Throughout, we refer to a copy-input baseline, defined as the trivial system that outputs the clean input unchanged; this baseline anchors all relative-improvement figures reported in evaluation.

In the U-Net, each of the ten effect labels receives a learned 64-dimensional embedding, injected at three points. At the bottleneck the embedding is projected to 64 channels, broadcast, concatenated, and fused by a DoubleConv. At each of the four decoder levels, feature-wise linear modulation (FiLM) [17] maps the embedding to per-channel scale and shift parameters applied as $x \cdot (1+\gamma) + \beta$, where the $(1+\gamma)$ form is the identity at initialization; FiLM transfers cleanly to U-Nets [18]. Each encoder skip connection receives FiLM of the same form through its own per-level heads, so no path through the network bypasses the condition. Conditioning dropout zeroes the entire embedding with probability 0.15 during training, inspired by classifier-free guidance [19], so the network cannot learn to ignore its condition. The two VAE baselines condition on the target-effect index through a separate four-dimensional learned embedding, concatenated to the latent code before decoding (64-dimensional in the fully connected VAE, 512-dimensional in the ConvVAE). Conditioning therefore enters at a single point, with no FiLM, skip-connection modulation, or conditioning dropout.

The U-Net variants optimize a composite spectral loss whose base term is spectral convergence (SC), $\|\hat{Y} - Y\|_F / (\|Y\|_F + 1.0)$, where the large epsilon of 1.0 keeps the ratio bounded on near-silent targets. For the log-magnitude variant, the loss is SC + $10\,\text{L1}_{\text{log}}$, with the L1 term computed directly in the network's native log-magnitude domain. For the linear variant, the loss is SC + $10\,\text{L1}_{\text{lin}} + 5\,\text{L1}_{\text{log}}$: an L1 term in the native linear-magnitude domain, plus an auxiliary L1 term computed on a log-compressed version of the same prediction, included to penalize errors in low-energy regions that the linear term underweights. A per-effect weight $w = \min((\Delta_{\max}/\Delta_e)^{0.5}, 5.0)$ is applied to each sample's loss based on its target effect, where Δ_e is the mean clean-to-effect L1 distance of that effect and $\Delta_{\max}$ the largest such value across effects; this upweights effects with naturally small clean-to-effect spectral differences, which would otherwise contribute a vanishingly small gradient signal. These objectives apply to the U-Net variants; the VAE objectives are given above.

We train the U-Nets with AdamW [20] (learning rate 3×10^{-4}, weight decay 1×10^{-4}), ReduceLROnPlateau (factor 0.5, patience 8, minimum 1×10^{-6}), gradient clipping at norm 1.0, batch size 32, a RandomSampler with replacement over 8000-sample epochs, and a 90/10 train/validation split. The VAE baselines are trained on the same data split, using Adam at 1×10^{-3} with mixed precision; the full pipeline is built on librosa [21] and PyTorch [22].

We evaluate the four trained models on 200 held-out paired samples (20 per effect), drawn from the same split. All metrics are computed on linear shared-max magnitude spectrograms. Per-

Table 2: Per-effect breakdown of MSE and FAD improvements corresponding to Table 1, grouped by effect category. Unprocessed input columns show raw baseline values for each metric (lower is better); note that RAT has a baseline MSE an order of magnitude larger than all other effects, making it the easiest target; a pooled aggregate would be dominated by RAT, which is why Table 1 reports improvement averaged over effects. ΔMSE and ΔFAD give percentage improvement over the unprocessed input baseline (↑ better); "no imp" indicates performance below baseline. VAE is omitted: apart from a marginal +0.7% on RAT by MSE, it shows no improvement on any effect on either metric. Delay effects show notably weaker FAD gains despite strong MSE improvements, suggesting spectral error and perceptual quality diverge for time-based effects. Bold marks the best ΔMSE and the best ΔFAD in each row among the models.

	Unprocessed input		ConvVAE		U-Net ($\mathcal{L}_{\text{lin}}$)		U-Net ($\mathcal{L}_{\log}$)	
Effect	↓ MSE$_{\times 10^{-4}}$	↓ FAD	ΔMSE	ΔFAD	ΔMSE	ΔFAD	ΔMSE	ΔFAD
Distortion								
RAT	32.16	23.84	+65.9%	+64.3%	**+85.0%**	+78.3%	+84.8%	**+82.5%**
Tube Screamer	3.42	10.96	no imp	+50.9%	**+82.8%**	+55.4%	+80.0%	**+57.0%**
Modulation								
Flanger	3.39	9.56	no imp	no imp	+56.7%	+16.1%	**+57.2%**	**+28.3%**
Phaser	1.41	4.06	no imp	no imp	+66.9%	+21.7%	**+71.1%**	**+44.9%**
Delay								
Digital Delay	0.65	1.40	no imp	no imp	**+50.9%**	no imp	+49.9%	**+0.9%**
Sweep Echo	3.14	1.21	no imp	no imp	**+64.0%**	+25.9%	+63.5%	**+33.4%**
Tape Echo	1.63	2.08	no imp	no imp	+68.0%	**+25.7%**	**+74.0%**	+23.7%
Reverb								
Hall Reverb	1.90	6.54	no imp	no imp	+74.7%	+13.8%	**+76.4%**	**+24.2%**
Plate Reverb	1.20	3.29	no imp	no imp	+69.9%	+11.5%	**+70.3%**	**+23.1%**
Spring Reverb	2.71	5.26	no imp	no imp	+75.4%	no imp	**+78.6%**	**+0.9%**
Mean across effects								
All effects	5.16	6.82	no imp	no imp	+69.4%	+23.0%	**+70.6%**	**+31.8%**

bin MSE is complemented by Fréchet Audio Distance (FAD) [23]. Baseline-relative improvement is $(b - m)/b$ for model value m against the copy-input baseline b, and the overall value reported is the mean of the ten per-effect means.

3. RESULTS

Table 1 shows that the U-Net (Log) is best on both metrics (+70.6% MSE, +31.8% FAD), the U-Net (Linear) is second (+69.4% and +23.0%), and both variational autoencoder baselines fail to beat the copy-input baseline on average, with no positive mean per-effect improvement on either metric.

Table 2 groups the effects into distortion, modulation, delay, and reverb. The U-Net (Log) is the strongest model in most cells, performing best for six effects by MSE and nine by FAD; the U-Net (Linear) takes the four remaining MSE cells and the single remaining FAD cell. The two distortions sit at the top of the baseline-relative ranking and the time-based effects at the bottom, an ordering that tracks the size of the copy-input baseline: a distortion leaves little of the clean signal intact and so leaves the most error to remove, whereas a delay leaves the input largely unchanged. The two metrics diverge most on these time-based effects, which post strong MSE improvements but markedly weaker FAD gains; under the U-Net (Log), for example, Spring Reverb improves +78.6% on MSE but only +0.9% on FAD.

4. DISCUSSION

The strongest pattern in Table 2 is architectural since the U-Net's skip connections pass each encoder feature map directly to the corresponding decoder level. This likely preserves harmonic structure at every resolution, while the VAE family compresses the input through a latent bottleneck and loses fine detail. This is a previously documented source of blurred VAE reconstructions [24]. RAT is the exception, the only case where ConvVAE clears the baseline on both metrics. The copy-input baseline is the weakest, so capturing the broad spectral envelope is enough to beat it.

We use effect macro-averaging because MSE on a magnitude spectrogram is scale-dependent: RAT's copy-input baseline is an order of magnitude above every other effect, so it looks hardest by absolute MSE yet is the easiest in relative terms, since the unchanged input leaves the most error to remove. Effect-averaging keeps it from dominating the aggregate: the ConvVAE, for instance, has a lower pooled MSE than the baseline (4.09 vs 5.16×10^{-4}) but clears the baseline on no effect except RAT, so its effect-averaged improvement is not positive. Log compression helps because it lifts low-amplitude detail such as reverb tails, modulation sidebands, and delay echoes into a range the encoder features and the loss can represent. The resulting MSE gap between the two U-Nets is small (1.2 percentage points) but the FAD gap is larger (8.8 percentage points) and consistent (nine out of all effects), in line with input representation mattering for audio networks [25].

As an informal diagnostic, we examine whether the U-Net responds to the conditioning label rather than collapsing toward a single transformation. For each clean input, we generate outputs for all effect labels and compute the mean pairwise L1 distance between predictions, normalized by the mean L1 distance between each prediction and the unprocessed input (Table 3). Ratios substantially above zero indicate label-dependent variation. All effects produce ratios above 1.0 (mean 1.416), providing qualitative evidence that the conditioning signal influences the generated transformations rather than being ignored by the model. This diag-

Table 3: Mean pairwise L1 distance between predictions obtained by querying the trained model with all effect labels for a fixed clean input, divided by the mean L1 distance between each such prediction and the unprocessed input. A ratio near zero would indicate that predictions do not vary appreciably with the conditioning label. This table is reported as an informal check of the best U-Net model not collapsing.

Effect	Mean ratio
RAT	1.458
Tube Screamer	1.489
Flanger	1.463
Phaser	1.432
Digital Delay	1.431
Sweep Echo	1.422
Tape Echo	1.475
Hall Reverb	1.132
Plate Reverb	1.462
Spring Reverb	1.397
Mean across effects	
All effects	1.416

nostic should be interpreted only as evidence against conditioning collapse, not as validation of effect-specific generation.

The demo website probes out-of-distribution behavior, pairing four EGFxSet-derived passages with Guitar-TECHS [11] real performance recordings, each processed by the log-magnitude U-Net and the ConvVAE under all effects. On Guitar-TECHS input the U-Net produces audible but weaker transformations, and the ConvVAE shows the same smoothed-upper-band failure seen in-distribution. The demo therefore contrasts with the EGFxSet-derived distribution, within which the headline results hold.

5. REFERENCES

[1] D. Hunter, *Guitar Effects Pedals: The Practical Handbook*. Backbeat Books, 2004.

[2] J. Ostberg and B. J. Hartmann, "The electric guitar–marketplace icon," *Consumption Markets & Culture*, vol. 18, no. 5, pp. 402–410, 2015.

[3] J. Schneider, *The Contemporary Guitar*. University of California Press, 1985, vol. 5.

[4] T. Wilmering, D. Moffat, A. Milo, and M. B. Sandler, "A history of audio effects," *Applied Sciences*, vol. 10, no. 3, p. 791, 2020.

[5] J. O. Smith, *Introduction to digital filters: with audio applications*. Julius Smith, 2007.

[6] M. A. Martínez Ramírez, E. Benetos, and J. D. Reiss, "Deep learning for black-box modeling of audio effects," *Applied Sciences*, vol. 10, no. 2, p. 638, 2020.

[7] A. Wright, E.-P. Damskägg, L. Juvela, and V. Välimäki, "Real-time guitar amplifier emulation with deep learning," *Applied Sciences*, vol. 10, no. 3, p. 766, 2020.

[8] C. J. Steinmetz, N. J. Bryan, and J. D. Reiss, "Style transfer of audio effects with differentiable signal processing," *J. Audio Eng. Soc.*, 2022.

[9] H. Pedroza, G. Meza, and I. R. Roman, "EGFxSet: Electric guitar tones processed through real effects of distortion, modulation, delay and reverb," in *International Society for Music Information Retrieval Conference*, Bengaluru, India, 2022.

[10] H. Pedroza, W. Abreu, R. M. Corey, and I. R. Roman, "Leveraging real electric guitar tones and effects to improve robustness in guitar tablature transcription modeling," in *Proceedings of the 27th International Conference on Digital Audio Effects (DAFx24)*, 2024, pp. 143–146.

[11] H. Pedroza *et al.*, "Guitar-TECHS: An electric guitar dataset covering techniques, musical excerpts, chords and scales using a diverse array of hardware," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2025.

[12] D. W. Griffin and J. S. Lim, "Signal estimation from modified short-time Fourier transform," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 236–243, 1984.

[13] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015, pp. 234–241.

[14] A. Jansson *et al.*, "Singing voice separation with deep U-Net convolutional networks," in *Proc. Int. Society for Music Information Retrieval Conf. (ISMIR)*, 2017, pp. 745–751.

[15] Y. Wu and K. He, "Group normalization," in *Proc. European Conf. Computer Vision (ECCV)*, 2018, pp. 3–19.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[17] E. Perez *et al.*, "FiLM: Visual reasoning with a general conditioning layer," in *AAAI Conf. Artificial Intelligence*, 2018.

[18] G. Meseguer-Brocal and G. Peeters, "Conditioned-U-Net: Introducing a control mechanism in the U-Net for multiple source separations," in *Proc. Int. Society for Music Information Retrieval Conf. (ISMIR)*, 2019, pp. 159–165.

[19] J. Ho and T. Salimans, "Classifier-free diffusion guidance," in *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*.

[20] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2019.

[21] B. McFee, M. McVicar, D. Faronbi, I. Roman, M. Gover, S. Balke, S. Seyfarth, A. Malek, C. Raffel, V. Lostanlen *et al.*, "librosa/librosa: 0.10. 0," *zenodo*.

[22] A. Paszke *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[23] K. Kilgour, M. Zuluaga, D. Roblek, and M. Sharifi, "Fréchet audio distance: A reference-free metric for evaluating music enhancement algorithms," in *Proc. Interspeech*, 2019, pp. 2350–2354.

[24] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," in *International conference on machine learning*. PMLR, 2016, pp. 1558–1566.

[25] K. W. Cheuk, K. Agres, and D. Herremans, "The impact of audio input representations on neural network based music transcription," in *Proc. Int. Joint Conf. Neural Networks (IJCNN)*, 2020, pp. 1–8.