OFF-MANIFOLD ROBUSTNESS IN SYNTHESIZER INVERSION WITH JOINT DISTRIBUTION FLOW MATCHING (SUPPLEMENTARY MATERIAL)

Anonymous Authors
Anonymous Affiliations
anonymous@ismir.net

A note to reviewers: This supplementary material is intended to serve as an optional addition to the paper. The information contained within it is included to aid reproducibility by providing further detail on training configurations and architectural choices. None of it is necessary in order to understand the contributions of the main paper, and reviewers should not feel obliged to read it. We offer it for completeness, and so reviewers seeking clarifying information are able to access it.

On submission of the final version, we intend to make this supplementary material available to readers on the companion website. This note, of course, will be omitted.

1. TASK-MIXTURE SAMPLING

Table 1 gives the task-mixture weights used to sample μ for each model variant. Conditional and marginal distributions are achieved practically by fixing the time dimension corresponding to the conditioning or marginal variable to a constant value of 1 or 0, respectively, and sampling the remaining time variable from a logit-normal distribution [1].

JDF spreads training mass uniformly between the the joint and two conditional tasks. JDF-A introduces unconditional audio marginals, for both synthesised and real data. Unif-PARAMONLY and Conditional both model predominantly the conditional parameter distribution, with 10% of training samples devoted to a null-conditioning path to enable classifier-free guidance.

For audio-marginal tasks, no parameter target is observed and the parameter velocity loss is omitted.

Model	Task	Weight
JDF	Joint	$1/3$
	$p(\mathbf{x} \mid \mathbf{y})$	$1/3$
	$p(\mathbf{y} \mid \mathbf{x})$	$1/3$
JDF-A	Joint	$1/4$
	$p(\mathbf{x} \mid \mathbf{y})$	$1/4$
	$p(\mathbf{y} \mid \mathbf{x})$	$1/6$
	$p_{\text{synth}}(\mathbf{y})$	$1/6$
	$p_{\text{real}}(\mathbf{y})$	$1/6$
Unif-PARAMONLY	$p(\mathbf{x} \mid \mathbf{y})$	0.9
	$p(\mathbf{x})$	0.1
Conditional	$p(\mathbf{x} \mid \mathbf{y})$	0.9
	$p(\mathbf{x})$	0.1

Table 1. Task-mixture weights used to define μ during training.

Task	Initial point	Final point
Joint	$(0, 0)$	$(1, 1)$
$p(\mathbf{x} \mid \mathbf{y})$	$(0, 1)$	$(1, 1)$
$p(\mathbf{y} \mid \mathbf{x})$	$(1, 0)$	$(1, 1)$
$p(\mathbf{y})$	$(0, 0)$	$(0, 1)$
$p(\mathbf{x})$	$(0, 0)$	$(1, 0)$

Table 2. Inference-time paths through the time square.

2. SAMPLING PATHS

Table 2 defines the paths through the time square used for each task. For paths with one fixed modality, the corresponding clean reference is held fixed while the other modality is integrated from noise to data.

For noisy conditioning, the reference audio $\mathbf{y}^*$ is replaced by

$$Y_\tau = (1 - \tau)Y_0 + \tau\mathbf{y}^*, \tag{1}$$

and inversion is performed along the horizontal path from $(0, \tau)$ to $(1, \tau)$.

3. ARCHITECTURE DETAILS

All variants share a common spectrogram front-end and a Diffusion Transformer (DiT) [2] backbone, but differ in how the audio modality is integrated.

3.1 Tokenisation

3.1.1 Audio

Audio signals are first converted to a log-mel spectrogram (with Slaney scale and normalisation). The sample rate $f_s = 44.1$ kHz, $n_{\text{mels}} = 128$, window 25 ms, hop 10 ms (= 100 fps). For $T = 4$ s of audio this yields a tensor of shape $(2, 128, 401)$.

Both architectures use the same patch tokenizer. Channels and mel bins are flattened into a single feature axis of size $2 \cdot 128 = 256$, and a 1-D convolution with kernel size 2 and stride 2 over the time axis produces $\lceil 401/2 \rceil = 201$ tokens. Each token is then refined by a residual MLP and added to a learned absolute time-position embedding. We deliberately keep the full mel spectrum on the feature axis, rather than 2-D patchifying over frequency and time, so that every token captures the entire spectral content at its time slice. [3] Spectrograms statistics are standardised by an online Welford estimator that is frozen after $8,000$ training steps before patchification.

3.1.2 Synthesizer Parameters

A synthesizer configuration is represented as a mixed continuous–categorical vector. Continuous parameters are stored as scalars in $[0, 1]$, while categorical parameters are stored as one-hot vectors. We additionally model two note-level variables: MIDI pitch, treated as a discrete value in $[36, 84]$, and note duration, treated as a continuous value in $[1, 4]$ seconds. Both are rescaled to the model's input range and appended to the parameter vector.

We tokenize this vector using a feature-tokenization scheme for tabular data [4]. Let N denote the number of synthesizer parameters after grouping each scalar parameter or one-hot block as a single feature, and let d_{vec} be the length of the full concatenated vector. The vector is first multiplied element-wise by a learnable weight tensor $W \in \mathbb{R}^{d_{\text{vec}} \times d_{\text{model}}}$. For each parameter, we then sum the weighted contributions over the corresponding scalar entry or one-hot block, producing one token in $\mathbb{R}^{d_{\text{model}}}$. Finally, we add a learnable parameter-specific bias $b \in \mathbb{R}^{N \times d_{\text{model}}}$, which gives each token a distinct identity in the residual stream.

The inverse projection follows the same grouping structure. Each parameter token is broadcast back to the scalar entry or one-hot block from which it was formed, then projected with a learned output matrix to reconstruct the original vector representation. This preserves an explicit one-token-per-parameter correspondence, making the representation easy to invert and interpret.

We use Surge XT in a no-effects, mostly-deterministic configuration with $N = 140$ synthesizer parameters, totaling $d_{\text{vec}} = 290$ dimensions when discrete parameters are one-hot encoded. For Dexed, $N = 105$ and $d_{\text{vec}} = 226$.

3.2 Joint Flow Models (JDF / JDF-A) and Unif-PARAMONLY

The joint-flow variants and Unif-PARAMONLY use a single multi-modal DiT operating on the concatenation of spectrogram and parameter tokens. The transformer has $L = 8$ blocks, $d_{\text{model}} = 768$, $h = 12$ attention heads and feed-forward width $d_{\text{ff}} = 1536$. Both modalities live on the same residual stream and share attention heads. Per-modality output projections produce the two velocity targets, one for the spectrogram and one for the parameter vector.

Each block receives a layer-wise conditioning vector formed by concatenating two sinusoidal time embeddings (dimension 256 each, one for $t_{\mathbf{x}}$ and one for $t_{\mathbf{y}}$) and projecting them through a 2-layer MLP to $L \times d_{\text{model}}$. The conditioning is injected via Adaptive Layer Normlization [2]. Spectrogram tokens additionally carry a learned 1-D positional encoding over time, and parameter tokens carry a learned per-parameter bias from the projection.

3.3 Conditional Model

The Conditional baseline replaces the joint flow with a more conventional encoder-decoder architecture, based on the model presented in [5]. An encoder based on the audio spectrogram transformer (AST) [6] processes the spectrogram into a fixed conditioning tensor, and a separate parameter-only DiT generates parameters via flow matching, conditioned on the encoder output.

The encoder has $L_{\text{enc}} = 8$ blocks, $d_{\text{model}}^{\text{enc}} = 512$, $h_{\text{enc}} = 8$ heads, gated self-attention [7], and RoPE [8] on the time axis. After the audio tokens we append $L_{\text{flow}} = 8$ learnable embedding tokens. The post-encoder values of those tokens are projected to $d_{\text{cond}} = 128$ and serve as one conditioning vector per layer of the parameter flow. This gives the encoder room to allocate distinct features to different flow depths rather than collapsing everything to a single global descriptor.

The parameter flow uses $L = 8$ DiT blocks, $d_{\text{model}} = 512$, $h = 8$ heads, $d_{\text{ff}} = 1024$, gated self-attention [7], and adaptive LayerNorm conditioned by (i) a sinusoidal $t_{\mathbf{x}}$ embedding and (ii) the per-layer encoder vector. Because the encoder runs once per audio clip and the velocity field network has no spectogram branch, the schedule-path and joint-generation experiments are not defined for this variant: there is no $t_{\mathbf{y}}$ to schedule.

4. TRAINING DETAILS

All variants are trained by flow matching [9] with deterministic linear paths for $500,000$ steps on synthetic audio rendered on the fly by the target VST plugin, except JDF-A which interleaves real audio. We use the Adam optimiser with learning rate 2×10^{-4}, a $2,000$-step linear warm-up, and cosine decay to $0.01 \times$ lr. Gradients are clipped to a maximum $\updownarrow_2$ norm of 1.0, and training is conducted *without* EMA. We observe a slight performance boost across models when using the mean absolute error in place of the mean squared error when computing the velocity field loss. A global batch size of 64 is used for all models, with $\frac{1}{6}$ of samples drawn from the off-manifold dataset in the case of JDF-A. Training is performed with mixed-precision (bfloat16) on a single 24 GB GPU. This takes approximately 24 h for the Conditional baseline and 36 h for the joint-flow models.

5. REFERENCES

[1] P. Esser, S. Kulal, A. Blattmann, R. Entezari, J. Müller, H. Saini, Y. Levi, D. Lorenz, A. Sauer, F. Boesel, D. Podell, T. Dockhorn, Z. English, and R. Rombach, "Scaling rectified flow transformers for high-resolution image synthesis," in *Proceedings of the 41st*

International Conference on Machine Learning, ser. ICML'24, vol. 235. Vienna, Austria: JMLR.org, Jul. 2024, pp. 12 606–12 633.

[2] W. Peebles and S. Xie, "Scalable Diffusion Models with Transformers," in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. Paris, France: IEEE, Oct. 2023, pp. 4172–4182.

[3] A. Makineni, B. Geng, and Q. Tian, "Full-Frequency Temporal Patching and Structured Masking for Enhanced Audio Classification," Aug. 2025, arXiv:2508.21243 [cs].

[4] Y. Gorishniy, I. Rubachev, V. Khrulkov, and A. Babenko, "Revisiting deep learning models for tabular data," in *Proceedings of the 35th International Conference on Neural Information Processing Systems*, ser. NIPS '21. Red Hook, NY, USA: Curran Associates Inc., Dec. 2021, pp. 18 932–18 943.

[5] B. Hayes, C. Saitis, and G. Fazekas, "Audio synthesizer inversion in symmetric parameter spaces with approximately equivariant flow matching," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference*, Daejeong, Korea, Jun. 2025.

[6] Y. Gong, Y.-A. Chung, and J. Glass, "AST: Audio Spectrogram Transformer," 2021, pp. 571–575.

[7] Z. Qiu, Z. Wang, B. Zheng, Z. Huang, K. Wen, S. Yang, R. Men, L. Yu, F. Huang, S. Huang, D. Liu, J. Zhou, and J. Lin, "Gated Attention for Large Language Models: Non-linearity, Sparsity, and Attention-Sink-Free," Oct. 2025.

[8] J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo, and Y. Liu, "RoFormer: Enhanced transformer with Rotary Position Embedding," *Neurocomput.*, vol. 568, no. C, Feb. 2024.

[9] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le, "Flow Matching for Generative Modeling," Feb. 2023.