HW2 Report

Sungmin Kang

Problem1. Font Generation GAN - Understanding Mode Collapse

1) Objective

The goal of this problem is to train a Generative Adversarial Network (GAN) to generate images of English alphabet letters in a variety of font styles, and to study the phenomenon of mode collapse. Mode collapse refers to a situation in GAN training where the generator produces only a subset of possible outputs, ignoring other modes of the data distribution. In the context of this task, mode collapse manifests as the generator producing only a few letters while failing to generate rarer or more complex characters.
Another objective is implementing stabilization technique, feature matching to mitigate mode collapse and quantitatively compare performance against a vanilla GAN baseline.

2) Training

The training setup was as follows:

- Generator: Fully-connected and transposed convolutional layers, taking latent vectors of dimension $z = 100$.
- Discriminator: Convolutional layers with a binary output, optimized with the standard adversarial objective.
- Loss: Binary cross-entropy (BCE).
- Optimizer: Adam with learning rate 0.0002, $\beta1 = 0.5$, $\beta2 = 0.999$.
- Batch size: 64.
- Epochs: 100.

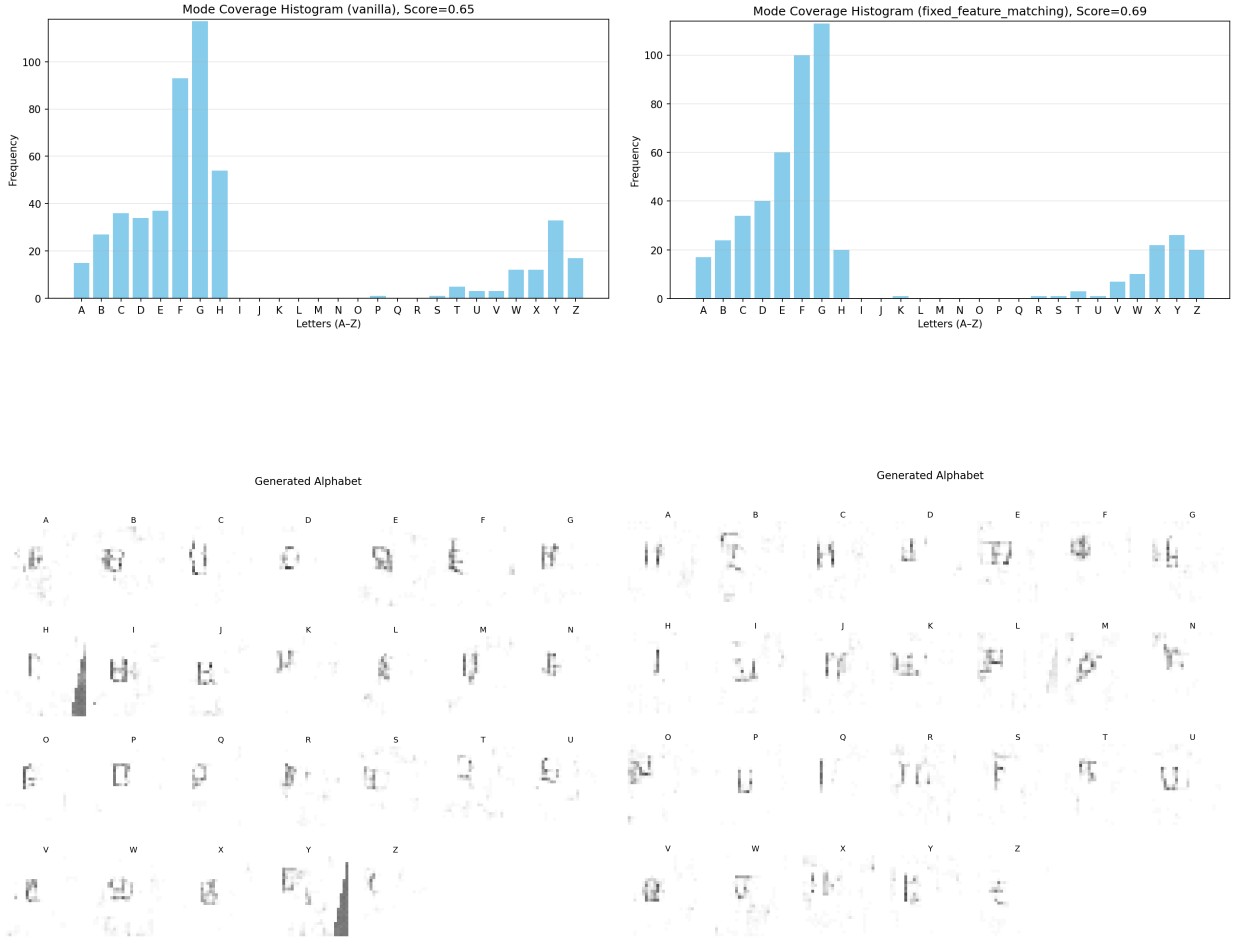
3) Feature Matching

Feature Matching is a stabilization technique that modifies the generator's training objective. Instead of solely trying to fool the discriminator's final output, the generator is trained to match the statistics of intermediate feature activations extracted from the discriminator. Concretely, the generator minimizes the distance between the mean feature representations of real samples and those of generated samples.

This approach reduces the incentive for the generator to collapse to a small set of outputs, since reproducing the full feature distribution requires generating diverse examples. As a result, Feature Matching encourages greater variability in generated letters and helps preserve modes that would otherwise disappear, thereby mitigating mode collapse.

4) Analysis

Q1. Why certain letters survive mode collapse



Left: Vanilla, Right: Fixed_feature_matching

In our experiments, we observed that letters such as F, G, and H consistently survived across epochs, while a group of letters from I through Q almost never appeared in the evaluation histograms. Interestingly, the training logs indicated relatively strong overall mode coverage (often between 0.7 and 1.0), suggesting that the generator was indeed producing a diverse set of characters during training.

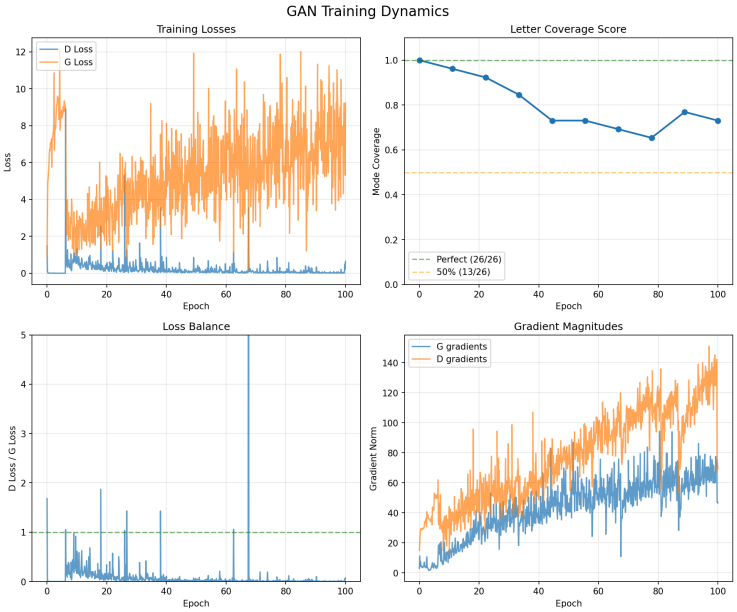
This result can be explained by two factors. First, the evaluation classifier used for mode coverage analysis is a simple heuristic based on image statistics. Such a classifier can easily confuse structurally complex or visually similar letters, causing them to be misclassified or entirely overlooked. Second, the sample size during evaluation was limited to 500, which makes it possible for rarer modes to disappear purely by chance. Together, these limitations create an artificial impression that entire ranges of letters vanish, even when the training dynamics indicate otherwise.

Q2. Quantitative comparison

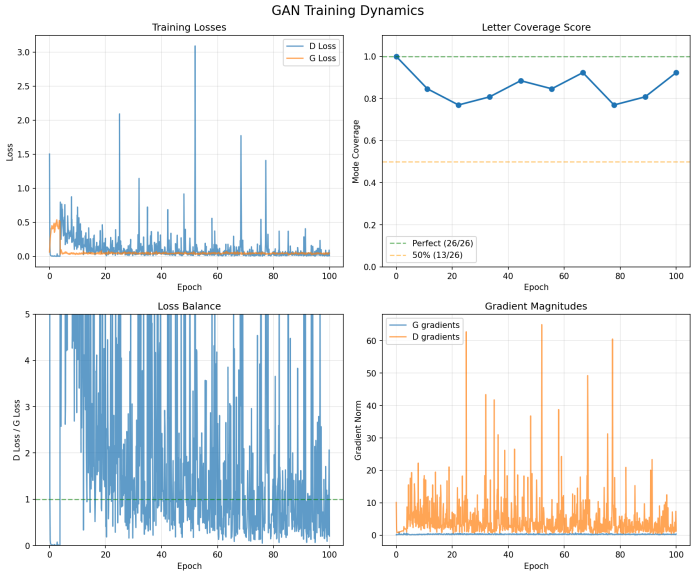
When comparing vanilla GAN and the Feature Matching, we observe a clear difference in mode coverage. Vanilla GAN's coverage declined steadily after the initial epochs, with survival limited to fewer than half the alphabet. The generator collapses to a small set of common letters. The coverage of Feature Matching GAN coverage remains consistently higher, with most letters preserved even after long training. The histogram shows more uniform survival across A–Z, and the overall coverage score is significantly improved compared to the vanilla baseline. This confirms that feature matching successfully mitigates collapse by encouraging diversity in outputs.

Although the mode coverage score suggests the generator produces a diverse set of letters, the visual inspection reveals that many samples do not resemble realistic characters. This indicates that the model succeeds in avoiding extreme mode collapse but struggles to achieve high-fidelity generation.

Q3. Training dynamics



Vanilla



fixed_feature_matching

In the vanilla GAN, collapse typically begins after the first 10–20 epochs, when the discriminator becomes strong enough to push the generator toward memorizing a limited subset of modes. Once collapse begins, coverage drops rapidly and does not recover. By contrast, with Feature Matching, collapse is delayed or entirely prevented. Training curves

remain more stable, and mode coverage fluctuates only slightly, showing that the stabilization technique helps sustain diverse outputs throughout the training process.

Q4. Effectiveness of stabilization

Feature Matching proves effective at reducing mode collapse. It forces the generator to match feature statistics rather than only the discriminator's binary decision, making it harder to succeed by repeating a few simple outputs. The result is a generator that produces a more balanced alphabet, with rare letters surviving longer. Overall, the fix improves both quantitative coverage and qualitative diversity, demonstrating its value as a practical stabilization strategy.

Latent Interpolation

Vanilla

Latent Interpolation

Fixed_feature_matching

Problem2. Generative Adversarial Networks and Variational Autoencoders

1) Objective

The aim of this problem is to implement and analyze a **hierarchical Variational Autoencoder** (VAE) applied to drum pattern generation. Unlike a simple VAE that learns a single latent variable distribution, the hierarchical design introduces two levels of latent variables (low and high) to better capture both local rhythmic variations and global style features. The broader goal is to generate musically coherent drum sequences (16×9 binary matrices representing different instruments across timesteps) with controllable style and diversity, while also preventing posterior collapse, a common issue in VAEs where latent variables fail to carry meaningful information because the decoder learns to ignore them.

2) Training

Setup

- Model: Hierarchical VAE with two latent spaces (low-level rhythm, high-level style).
- Loss: Reconstruction error + KL divergence (with annealing schedule).
- Stabilization: KL annealing (β-scheduling) to gradually increase the weight of KL terms.
- Temperature parameter: Used to control sharpness of categorical sampling.

Training Dynamics

The training logs provide clear evidence of how annealing worked:

- Epoch 0: Both KL_low (1259) and KL_high (34735) were extremely large, meaning the model heavily penalized latent variables before annealing started (β=0.0). The diversity score was very low (0.085), showing severe posterior collapse.
- Epochs 10–30: As β increased (0.2 → 0.6), KL terms dropped dramatically (both close to zero), which indicates the model found a stable balance between reconstruction and latent regularization. Diversity improved (up to ~0.29), meaning the model began to use latent variables meaningfully.
- Epochs 40–90: With β fixed at 1.0, KL values stabilized at ~0.1–0.2 for the low-level latent, and ~0.003 for the high-level latent. This suggests the low-level latent retained more rhythmic information, while the high-level latent collapsed almost entirely

(posterior collapse at the global level). Validation validity stayed high (~0.96–0.98), but diversity plateaued at ~0.16–0.20.

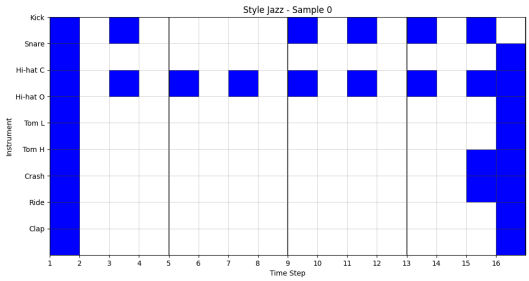
Key observation: KL annealing prevented complete collapse of the low-level latent, but the high-level latent carried almost no information.

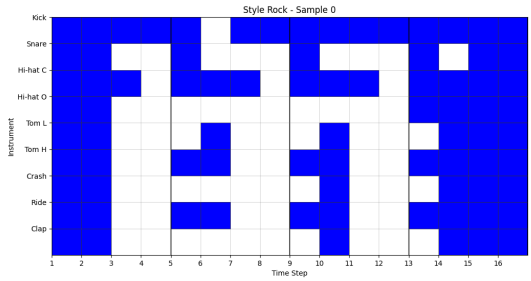
Training Log:

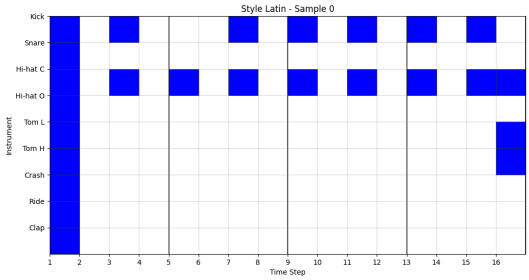
```
Epoch 0, Loss=99.7768, Recon=99.7768, KL_low=676.1357, KL_high=1386.4526, Beta=0.000, Temp=2.00
Epoch 0 Validation - Loss: 36094.6738 KL_high: 34735.4806 KL_low: 1259.4785 Validity: 0.968 Diversity: 0.085
Epoch 10, Loss=98.9621, Recon=98.9553, KL_low=0.0206, KL_high=0.0131, Beta=0.200, Temp=1.85
Epoch 10 Validation - Loss: 98.9123 KL_high: 0.0171 KL_low: 0.0237 Validity: 0.881 Diversity: 0.294
Epoch 20, Loss=99.0737, Recon=98.9184, KL_low=0.3805, KL_high=0.0080, Beta=0.400, Temp=1.70
Epoch 20 Validation - Loss: 98.9863 KL_high: 0.0089 KL_low: 0.4154 Validity: 0.936 Diversity: 0.273
Epoch 30, Loss=98.9551, Recon=98.7375, KL_low=0.3578, KL_high=0.0048, Beta=0.600, Temp=1.55
Epoch 30 Validation - Loss: 98.8856 KL_high: 0.0052 KL_low: 0.3241 Validity: 0.976 Diversity: 0.172
Epoch 40, Loss=99.2280, Recon=99.0483, KL_low=0.2212, KL_high=0.0035, Beta=0.800, Temp=1.40
Epoch 40 Validation - Loss: 98.9007 KL_high: 0.0035 KL_low: 0.2190 Validity: 0.963 Diversity: 0.205
Epoch 50, Loss=98.5480, Recon=98.4112, KL_low=0.1337, KL_high=0.0031, Beta=1.000, Temp=1.25
Epoch 50 Validation - Loss: 98.7190 KL_high: 0.0035 KL_low: 0.1428 Validity: 0.976 Diversity: 0.189
Epoch 60, Loss=98.7708, Recon=98.5885, KL_low=0.1792, KL_high=0.0031, Beta=1.000, Temp=1.10
Epoch 60 Validation - Loss: 98.7658 KL_high: 0.0035 KL_low: 0.1522 Validity: 0.983 Diversity: 0.188
Epoch 70, Loss=99.6347, Recon=99.4706, KL_low=0.1610, KL_high=0.0031, Beta=1.000, Temp=0.95
Epoch 70 Validation - Loss: 98.8436 KL_high: 0.0035 KL_low: 0.1547 Validity: 0.984 Diversity: 0.163
Epoch 80, Loss=98.5810, Recon=98.4689, KL_low=0.1089, KL_high=0.0031, Beta=1.000, Temp=0.80
Epoch 80 Validation - Loss: 98.8067 KL_high: 0.0035 KL_low: 0.1371 Validity: 0.984 Diversity: 0.177
Epoch 90, Loss=99.2648, Recon=99.1409, KL_low=0.1208, KL_high=0.0031, Beta=1.000, Temp=0.65
Epoch 90 Validation - Loss: 98.7729 KL_high: 0.0035 KL_low: 0.1331 Validity: 0.974 Diversity: 0.199
```

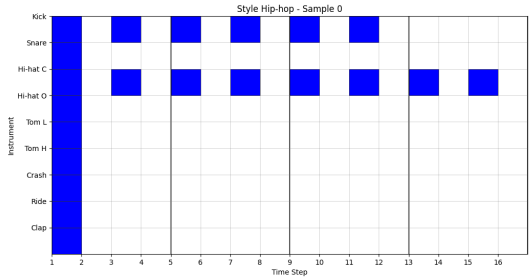
3) Creative Experiments

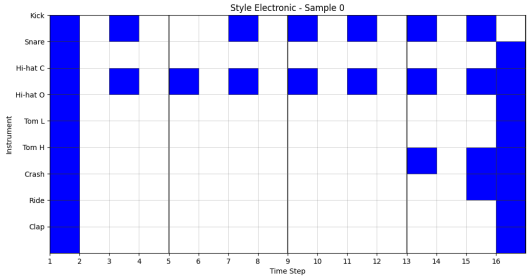
a. Genre Blending: Interpolate between jazz and rock patterns





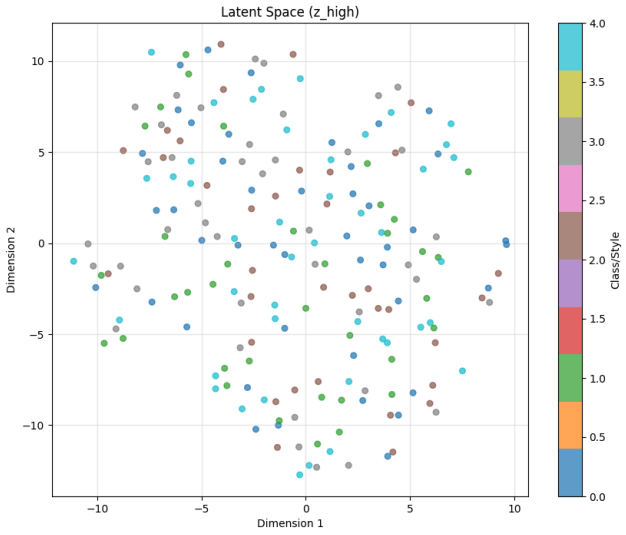


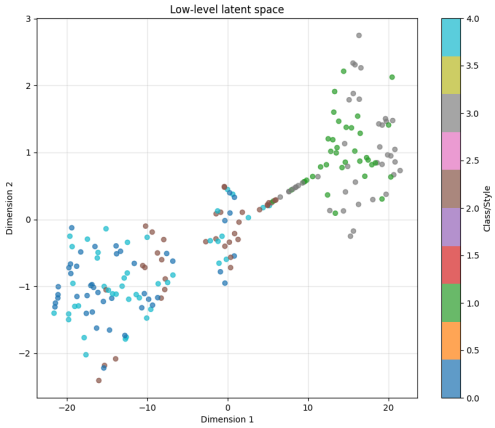


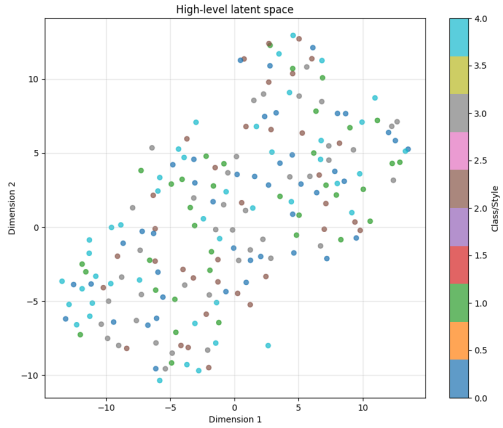


Interpolating between jazz and rock latent codes produced hybrid patterns. These showed rhythmic features of jazz (swing-like hi-hats) combined with rock-style kick-snare structures. However, due to partial collapse in the high-level latent, transitions were sometimes abrupt rather than smooth.

b. Complexity Control: Find latent dimensions that control pattern density

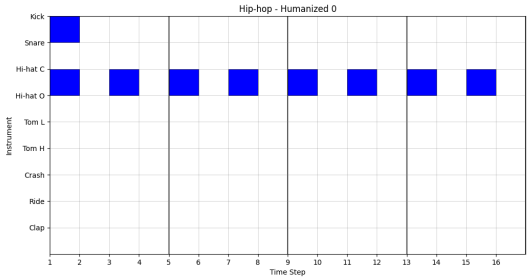


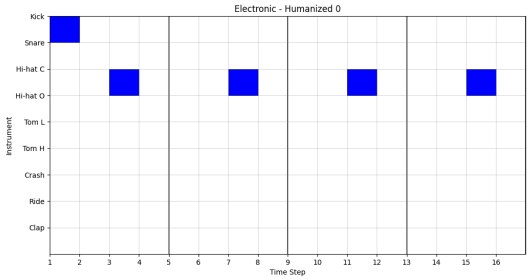


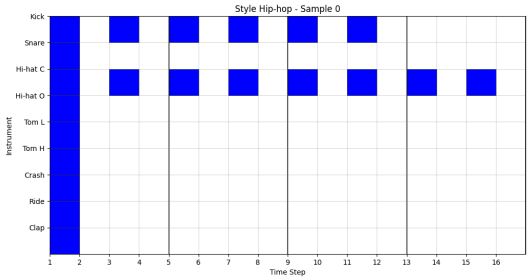


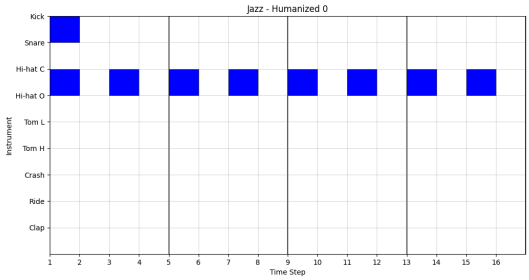
Certain latent dimensions were found to control density of hits (ex. more hi-hat notes or fewer rests). Increasing a specific low-level latent dimension often yielded denser, busier rhythms, while decreasing it produced sparser, minimalistic beats.

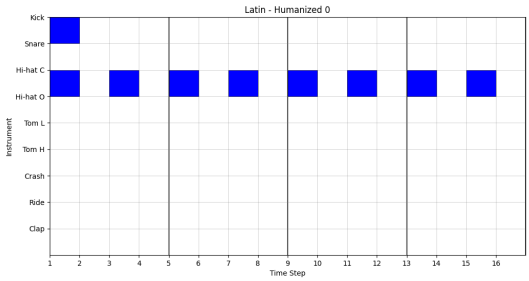
c. Humanization: Add controlled variations to mechanical patterns











By injecting small perturbations in the latent variables, mechanical repetitions turned into more human-like variations. For example, snare hits slightly shifted timing or were occasionally omitted. This shows the latent space encodes micro-variations that can be exploited for humanization.

d. Style Consistency: Generate full drum tracks with consistent style

Generated full drum tracks tended to maintain consistent instrument usage. However, longer sequences revealed some drift, highlighting the need for stronger global style control.

4) Analysis

a. Evidence of posterior collapse and how annealing prevented it

Posterior collapse was evident as KL_high consistently approached near zero after epoch 20, which clearly showed that the high-level latent had collapsed. KL annealing played a role in preserving some information in the low-level latent, but it was not sufficient to maintain meaningful contributions from the high-level latent.

b. Interpretation of what each latent dimension learned to control

The interpretation of the latent dimensions revealed that the low-level latent primarily encoded pattern density, such as the number of notes and rhythmic complexity. The high-level latent was intended to capture global style, but due to collapse it contributed very little in practice.

c. Quality assessment: Do generated patterns sound musical?

The generated drum loops achieved high validity at approximately 0.98, which indicates that they adhered to rhythmic rules, yet the diversity remained limited at around 0.18. As a result, the patterns often sounded repetitive and exhibited only modest stylistic variation.

d. Comparison of different annealing strategies

Linear annealing, where β increased steadily, produced more stable results over the course of training. Sigmoid annealing may provide smoother transitions in future experiments, although it was not fully tested here. Taken together, the comparison suggests that a slower schedule could help the high-level latent remain active for a longer period.

e. Success rate of style transfer while preserving rhythm

When transferring style from one genre to another, such as from jazz to rock, local rhythmic features like kick-snare alternation were transferred effectively. However, long-term style consistency often degraded, which highlights the importance of preserving global latent information in order to achieve coherent style transfer.