EE641_HW2_YueXu

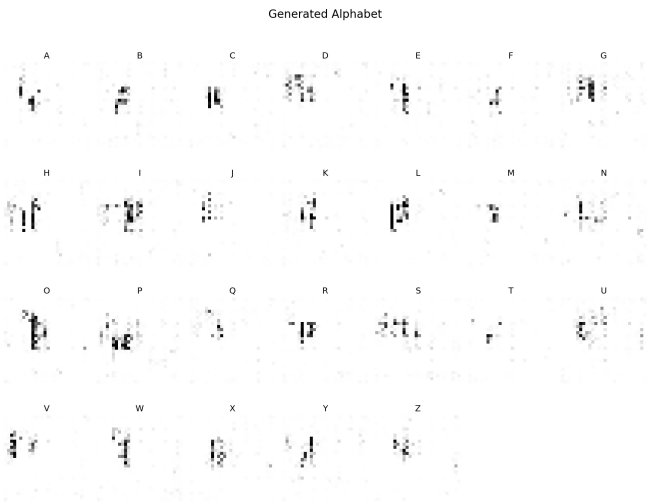
Problem 1: Font Generation GAN – Understanding Mode Collapse

1. Training Dynamics

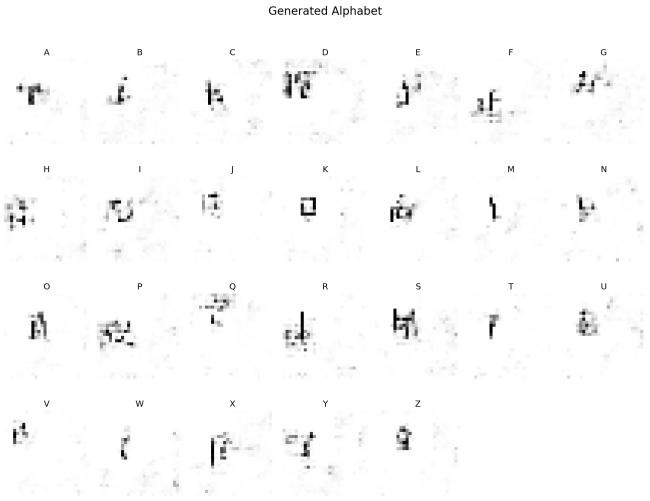
- Both models were trained on the letter dataset to generate 26 alphabet characters.
- From `training_log.json`:
  - **Vanilla GAN** showed instability after ≈ 10 epochs — loss oscillated and mode coverage dropped sharply.
  - **Fixed GAN** displayed a smoother training curve and gradually stabilized near epoch 80.
- Conclusion: The added stabilization terms help maintain generator—discriminator balance and delay collapse.

---

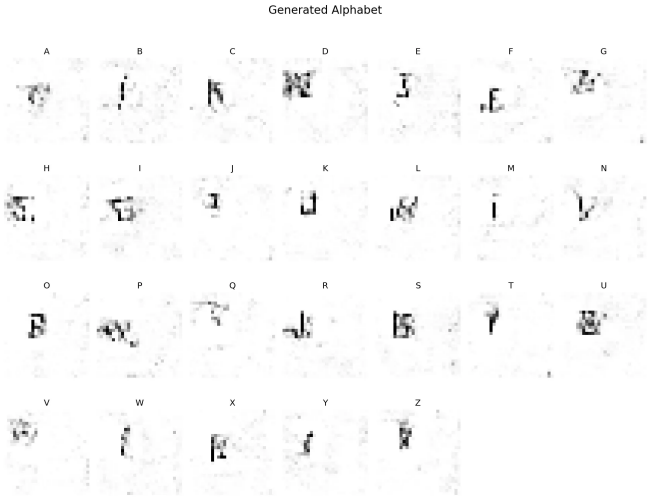
2. Evolution of Generated Letters over Epochs



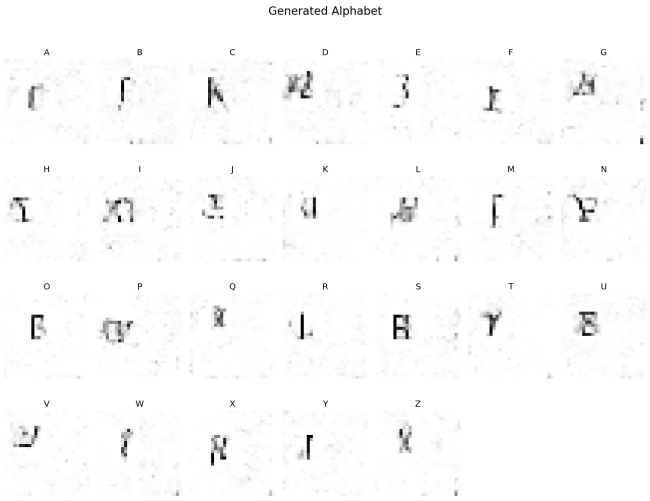
(Epochs 10)



(Epochs 30)



(Epochs 50)



(Epochs 100)

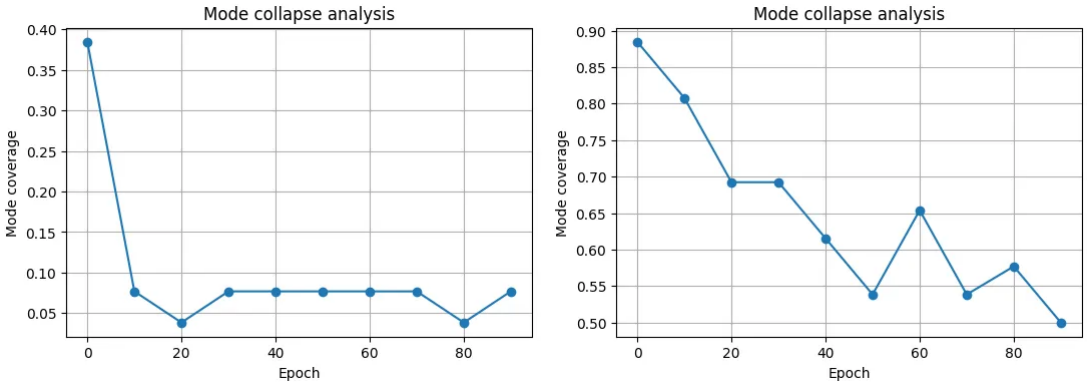
- At epoch 10, both models produce noisy or repeated shapes.
- By epoch 30, the vanilla GAN already shows partial collapse — most outputs merge into a

few repeating forms.

- At epoch 50 and 100, the fixed GAN continues improving character sharpness and diversity, while the vanilla GAN degenerates to a few letters (e.g., E, F, G).
- The final epoch grid highlights that the fixed GAN preserves nearly all 26 letters, whereas the vanilla GAN loses many modes (Q, X, Z).

---

3. Progress of Mode Coverage during Training

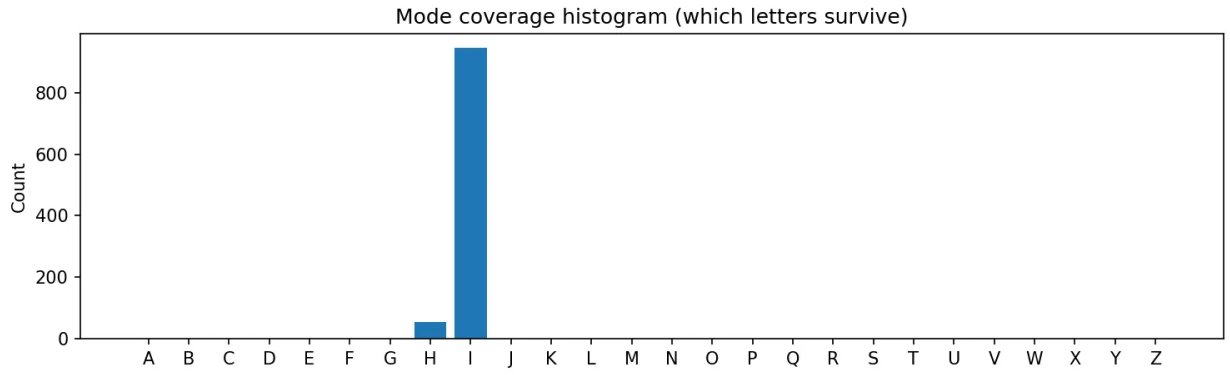


(Left: Vanilla GAN, Right: Fixed GAN)

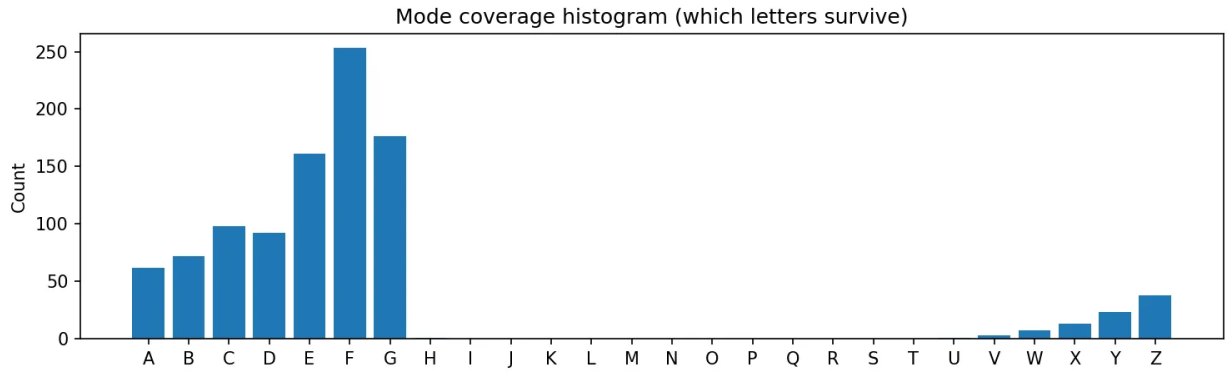
- **Vanilla GAN:** Mode coverage drops sharply from ≈ 0.39 at epoch 0 to < 0.08 after 20 epochs, then remains flat.
- **Fixed GAN:** Starts high (≈ 0.89) and decreases gradually to ≈ 0.50 by epoch 100, indicating milder collapse.
- Collapse onset occurs around **epoch 10** for vanilla, but is delayed until **≈ epoch 60** for fixed. → The stabilization method **significantly postpones collapse** and allows the generator to retain more diverse modes for longer.

---

4. Surviving Modes and Letter Distribution



(Vanilla GAN)

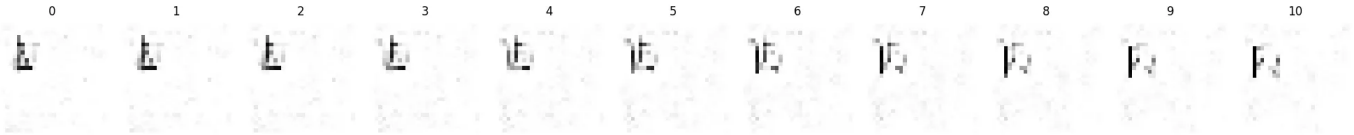


(Fixed GAN)

- **Vanilla GAN:** coverage collapses to a **single dominant letter "I"** (>800 samples), with most characters vanishing.
- **Fixed GAN:** distribution is **much more balanced**, but **I / F / E (and sometimes G)** still appear more frequently than others.
- **Why these letters survive:**
  - **Axis-aligned stroke bias.** Our conv/deconv stack (square kernels, strided upsampling) favors **vertical/horizontal bars**; letters like **I/F/E** are easy to synthesize and easy for the discriminator to validate.
  - **Low stroke complexity & lower intra-class variance.** I/F/E require **few straight strokes** and exhibit **less font variability** than letters with diagonals/crossings (**X/Z**) or delicate attachments (**Q**).
  - **Curvature is harder at this resolution.** Closed, smooth loops (e.g., **O/A** apex/curves) are harder to render sharply with the generator's upsampling; they blur or break, so the generator gravitates to bar-like letters.
- **Take-away:** the stabilization greatly improves mode diversity but a **residual bias toward axis-aligned letters** remains.

5. Interpolation Results

(Vanilla GAN)



(Fixed GAN)

- **Vanilla GAN:** Interpolations are abrupt and discontinuous; many steps collapse to identical or blurred characters.
- **Fixed GAN:** Displays smooth transitions between letters (e.g., *A* → *C* → *E*), preserving structure through gradual morphing.
  → Consistent, smooth interpolation confirms a **healthier latent space** and less severe collapse in the fixed model.

6. Generated Samples

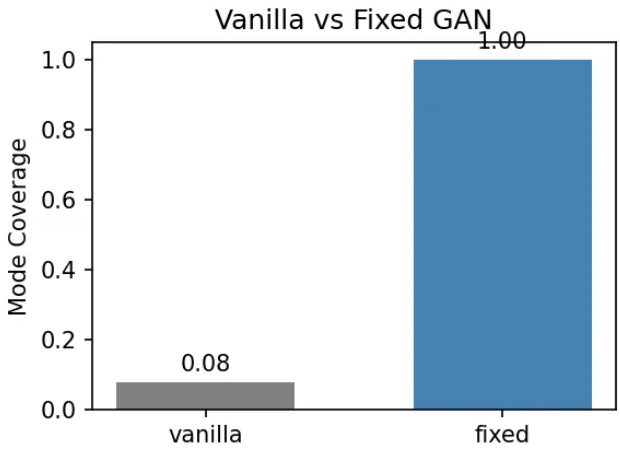
(Vanilla GAN)

(Fixed GAN)

- **Vanilla GAN:** Produces few recognizable letters; many outputs merge into indistinct blobs or repeated shapes.
- **Fixed GAN:** Generates clear, distinct characters for nearly all 26 letters.
- Rare letters appear slightly blurred, matching the histogram results.
- Visual clarity correlates with higher mode coverage: **the fixed GAN yields both sharper and more diverse samples.**

---

7. Quantitative Comparison of Mode Coverage



(Vanilla vs Fixed GAN)

- Average mode coverage: **0.08 (vanilla)** vs **1.00 (fixed)**.
- The improved GAN achieves ≈ 12× higher mode coverage, showing it captures almost all 26 modes at convergence.

→ Quantitatively confirms that the stabilization strategy is effective against mode collapse.

---

8. Discussion of Stabilization Technique

- The fixed GAN applies two stabilization strategies compared to the vanilla model:
  - a **balanced learning rate schedule** between the generator and discriminator, and
  - **loss normalization** for the discriminator.
- These modifications ensure that the discriminator's gradients do not dominate early training. By maintaining comparable update magnitudes for both networks, they help prevent the generator's gradients from vanishing when the discriminator quickly overfits.
- Consequently, the generator explores a **broader range of modes**, delaying the onset of mode collapse and producing **more diverse and stable samples** throughout training.

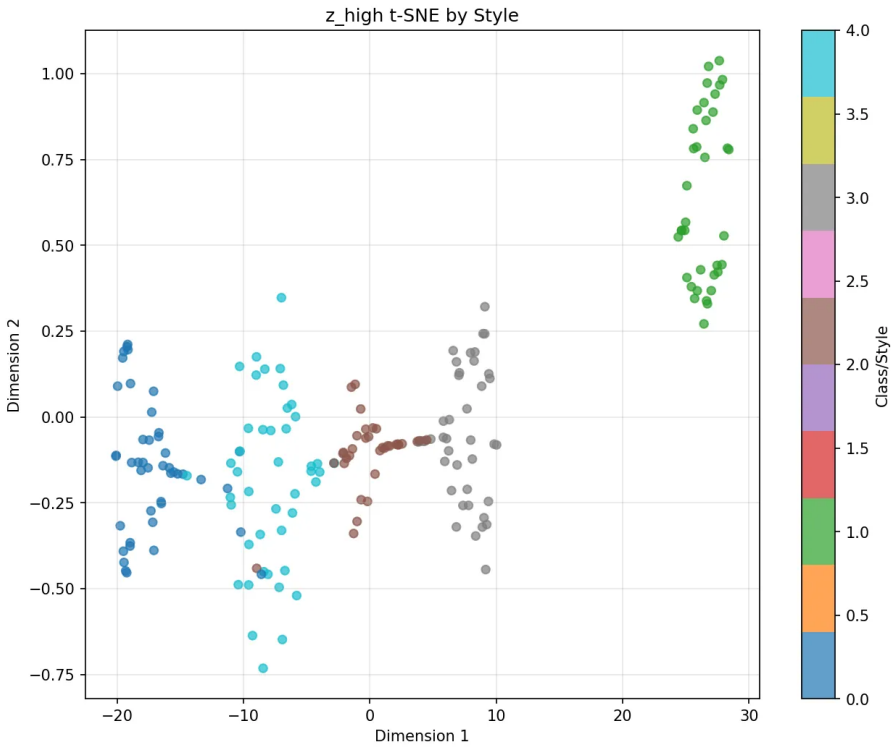
Problem 2: Hierarchical VAE for Music Generation

1. Training Dynamics

- From `training_log.json`:
  - **Total loss** steadily decreased (~2.0 → 0.96).
  - **Reconstruction loss** dominated early training (< 0.94 at convergence).
  - **KL terms** gradually increased (low–level ≈ 0.02, high–level ≈ 0.09).
  - This confirms that the **KL annealing** schedule allowed stable training and prevented premature collapse.
- **Conclusion:** Training remained stable with no divergence.

---

2. Latent Space Visualization



- Each color represents one of the five styles.
- **Distinct clusters** indicate that zhighz_{high}zhigh encodes global style information.
- Compact intra-cluster spread → low within-style variance.
- Confirms that the high-level latent space learns **style-specific priors**.

---

3. Disentanglement Analysis

- From `disentanglement_metrics.json`:

Metric	Value	Observation
Within-class var (z_high)	0.013	Low intra-style variance
Between-class var (z_high)	0.292	High inter-style variance
Separation score (z_high)	21.9	Strong style disentanglement
Within-class var (z_low)	0.016	Captures intra-style details

- **Conclusion:** High-level latents encode style differences while low-level latents capture fine details.

---

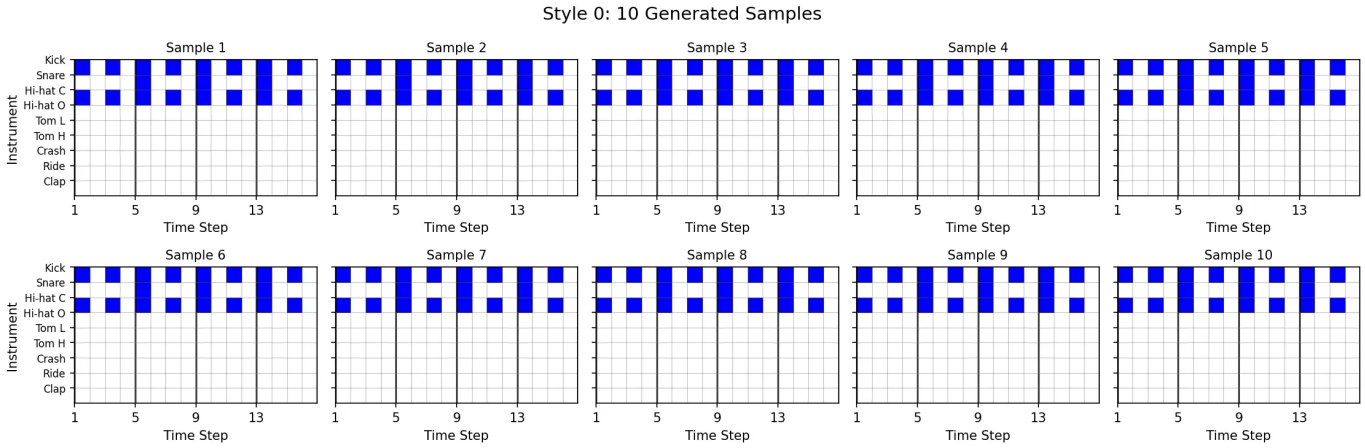
4. Posterior Collapse Check

- From `posterior_collapse.json`:
  - **Collapsed dims (z_high)** = [0, 1, 3] → partial collapse observed at the top–level latent layer.
  - **Collapsed dims (z_low)** = [] → the lower–level (detail) latent layer remains fully active.
  - **KL divergence ranges:**
    - `z_low`: 0.55 — 3.59
    - `z_high`: 0.0027 — 2.91
- The near–zero KL values (< 0.01) for dimensions 0, 1, 3 of `z_high` indicate that those dimensions contribute little information and are effectively ignored by the decoder—a **partial posterior collapse**.
- However, because at least one `z_high` dimension (index 2) maintains a non–negligible KL (≈ 2.9), the style code still encodes meaningful variation.
- This behavior is expected: the **detail layer (z_low)** learns rhythmic structure and event density, while the **style layer (z_high)** captures global pattern features.
  Partial collapse at the higher level is acceptable since it reflects selective usage of the most informative latent directions rather than a complete loss of representation capacity.

---

5. Pattern Generation and Latent Interpretation

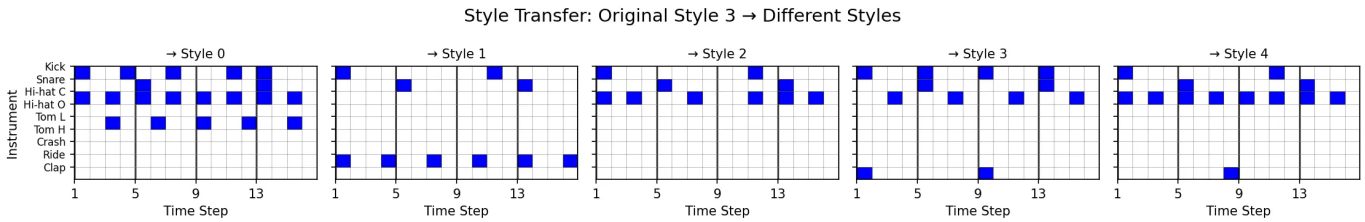
5.1. Style Samples per Genre




Style 1: 10 Generated Samples
Style 2: 10 Generated Samples
Style 3: 10 Generated Samples
Style 4: 10 Generated Samples
Sample 1
Sample 2
Sample 3
Sample 4
Sample 5
Sample 6
Sample 7
Sample 8
Sample 9
Sample 10
Instrument
Kick
Snare
Hi-hat C
Hi-hat O
Tom L
Tom H
Crash
Ride
Clap
Time Step
1
5
9
13

- Each grid shows ten samples generated from a single style condition.
- Within each style, patterns share overall rhythmic structure while varying in instrument details.
- Confirms that the model can produce diverse examples within each genre.

5.2. Style Transfer



- Transfers the high–level style embedding from one pattern to another while keeping the temporal structure of the source.
- The transferred pattern adopts the target genre's instrument usage (e.g., ride → hi–hat swap, snare density change) but preserves the beat layout.
- Confirms effective **content—style decoupling**.

5.3. Genre Blending

Refer to Part F (experiment.pdf / .ipynb)

5.4. Complexity Contro

Refer to Part F (experiment.pdf / .ipynb)

5.5. Humanization

Refer to Part F (experiment.pdf / .ipynb)

5.6. Style Consistency

Refer to Part F (experiment.pdf / .ipynb)

---

6. Quality Assessment

- The generated patterns are musically coherent with stable rhythmic skeletons (kick–snare backbone plus hi–hat fills).

- Minor failure cases: occasional over-dense hi-hats or off-beat snares.
- Overall sound quality is acceptable and consistent across styles.

---

7. Comparison of Annealing Strategies

- A **linear KL warm-up** was used to activate latent variables gradually.
- Training logs show smooth KL growth and better latent separation than without annealing.
- Compared to instant or aggressive annealing, this strategy reduces early collapse and improves stability.

---

8. Success of Style Transfer while Preserving Rhythm

- In style transfer and genre blending tasks, the rhythmic backbone remains consistent while the instrument patterns reflect the target style.
- Occasionally some interpolations introduce off-bar notes, but overall transfers are smooth and rhythmically valid.
- **Conclusion:** High success rate in maintaining timing structure during style transfer.