A. DDPM TRAINING AND SAMPLING

To train and sample from our diffusion model, we use the algorithms as described in [8].

Algorithm 2 Training

Input: $q(x_0)$, N steps, noise schedule $\beta_1, ..., \beta_N$
repeat
  $x_0 \sim q(x_0)$
  $t \sim \mathbb{U}(\{1, ..., N\})$
  $\sqrt{\bar{\alpha}} \sim \mathbb{U}(\sqrt{\bar{\alpha}_{t-1}}, \sqrt{\bar{\alpha}_t})$
  $\epsilon \sim \mathcal{N}(0, I)$
  Take gradient descent step on
  $\nabla_\theta \left\| \epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, \sqrt{\bar{\alpha}}) \right\|^2$
until converged

Algorithm 3 Sampling

Input: N steps, noise schedule $\beta_1, ..., \beta_N$
$x_N \sim \mathcal{N}(0, I)$
for $t = N, ..., 1$ **do**
  $\epsilon \sim \mathcal{N}(0, I)$ if $t > 1$, else $\epsilon = 0$
  $x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(x_t, \sqrt{\bar{\alpha}_t}) \right) + \sigma_t \epsilon$
end for
return x_0

B. MUSICVAE ARCHITECTURE

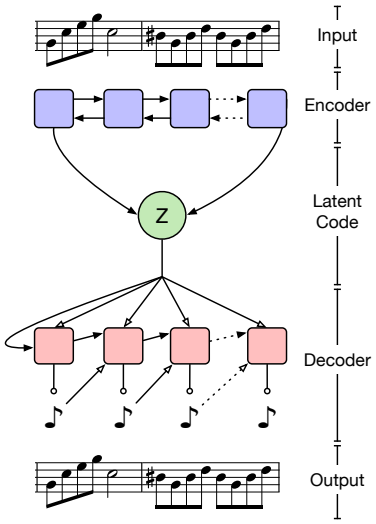

Figure 4. 2-bar melody MusicVAE architecture. The encoder is a bi-direction LSTM and the decoder is an autoregressive LSTM.

C. TRIMMING LATENTS

During training VAEs typically learn to only utilize a fraction of their latent dimensions. As shown in Figure 5, by examining the standard deviation per dimension of the posterior $q(z|y)$ averaged across the entire training set, we are able to identify underutilized dimensions where the average embedding standard deviation is close to the prior of 1. The VAE loss encourages the marginal posterior to match to the prior [42,43], but to encode information, dimensions must have smaller variance per an example.

In all experiments, we remove all dimensions except for the 42 dimensions with standard deviations below 1.0, before training the diffusion model on the input data. We find this latent trimming to be essential for training as it helps to avoid modeling unnecessary high-dimensional noise and is very similar to the distance penalty described in [4]. We also tried reducing the dimensionality of embeddings with principal component analysis (PCA) but found that the lower dimensional representation captured too many of the noisy dimensions and not those with high utilization.

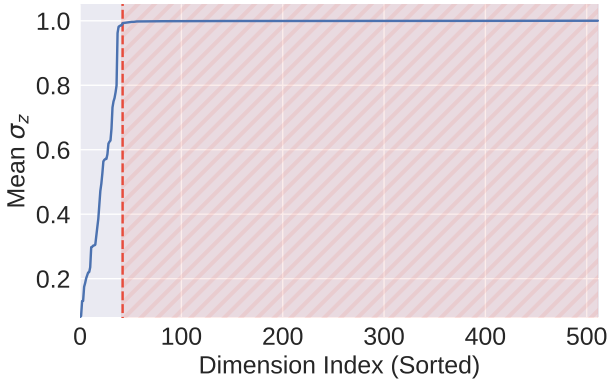

Figure 5. The standard deviation per dimension of the MusicVAE posterior $q(z|y)$ averaged across the entire training set. The region highlighted in red contains the latent dimensions that are unused.

D. TABLES

In Tables 2 and 3, we present the unnormalized framewise self-similarity results as well as the latent space evaluation of each model.

Setting	Unconditional				Infilling			
Quantity	Pitch		Duration		Pitch		Duration	
OA	μ	σ^2	μ	σ^2	μ	σ^2	μ	σ^2
Train Data	0.82	0.018	0.88	0.012	0.82	0.018	0.88	0.012
Test Data	0.82	0.018	0.88	0.011	0.82	0.018	0.88	0.011
Diffusion	**0.81**	**0.017**	**0.85**	**0.013**	**0.80**	**0.021**	**0.86**	**0.015**
Autoregression	0.76	0.024	0.82	0.015	-	-	-	-
Interpolation	0.94	0.004	0.96	0.004	0.87	0.014	0.91	**0.009**
$\mathcal{N}(0, I)$ Prior	0.69	0.033	0.79	0.016	0.73	0.033	0.82	0.018

Table 2. Unnormalized framewise self-similarity (overlapping area) evaluation of unconditional and conditional samples. Evaluations of same samples as in Table 1. Note the interpolations have unrealistically high mean overlap and low variance, while the Gaussian prior and TransformerMDN samples suffer from unrealistically lower mean overlap and higher variance.

Setting	Unconditional		Infilling	
Metric	FD$\times 10^{-2}$	MMD$\times 10^{-2}$	FD$\times 10^{-2}$	MMD$\times 10^{-2}$
Train Data	0.00	0.00	0.00	0.00
Test Data	1.24	0.12	1.24	0.12
Diffusion	1.66	0.18	1.53	0.16
Autoregression	**1.26**	**0.12**	-	-
Interpolation	3.22	0.43	1.97	0.23
$\mathcal{N}(0, I)$ Prior	2.44	0.29	**1.17**	**0.12**

Table 3. Latent space evaluation of infilling and unconditional and conditional samples. As described in Section 4.5, the TransformerMDN performs better in latent space similarity, even while producing less realistic samples (as seen in Tables 1 and 2).

E. ADDITIONAL SAMPLES

In Figure 6 we provide piano rolls of sequences drawn from the test set and in Figures 7, 8, 9, and 10 we present additional samples unconditionally generated by our diffusion model, TransformerMDN, spherical interpolation, and through independent sampling from the MusicVAE prior, respectively. Additional piano roll visualizations from infilling experiments are provided in Figure 11.

For extended visual and audio samples of the generated sequences from each model, we refer the reader to the online supplement available at `https://goo.gl/magenta/symbolic-music-diffusion-examples`.

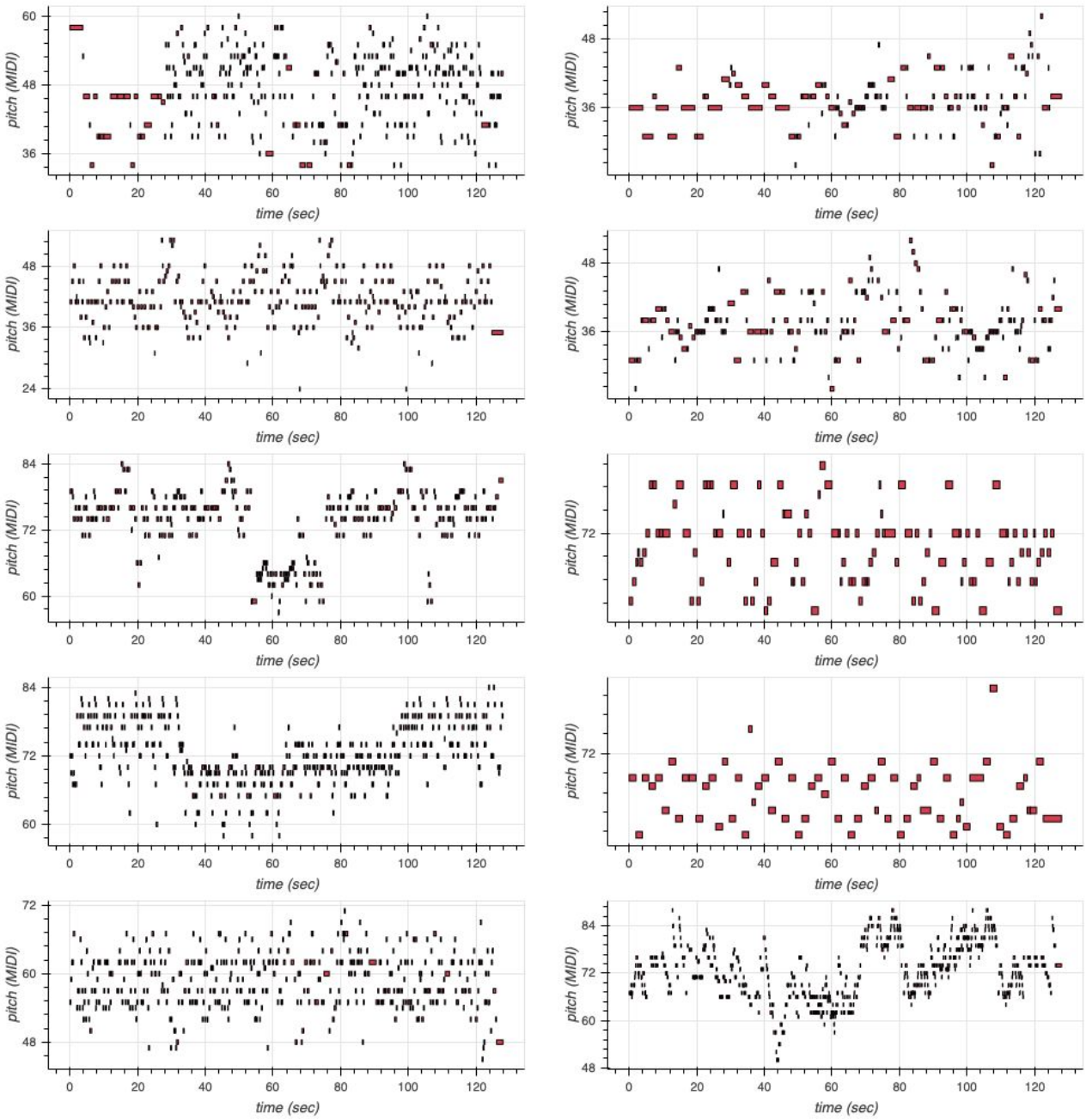

Figure 6. Additional piano rolls from the test set.

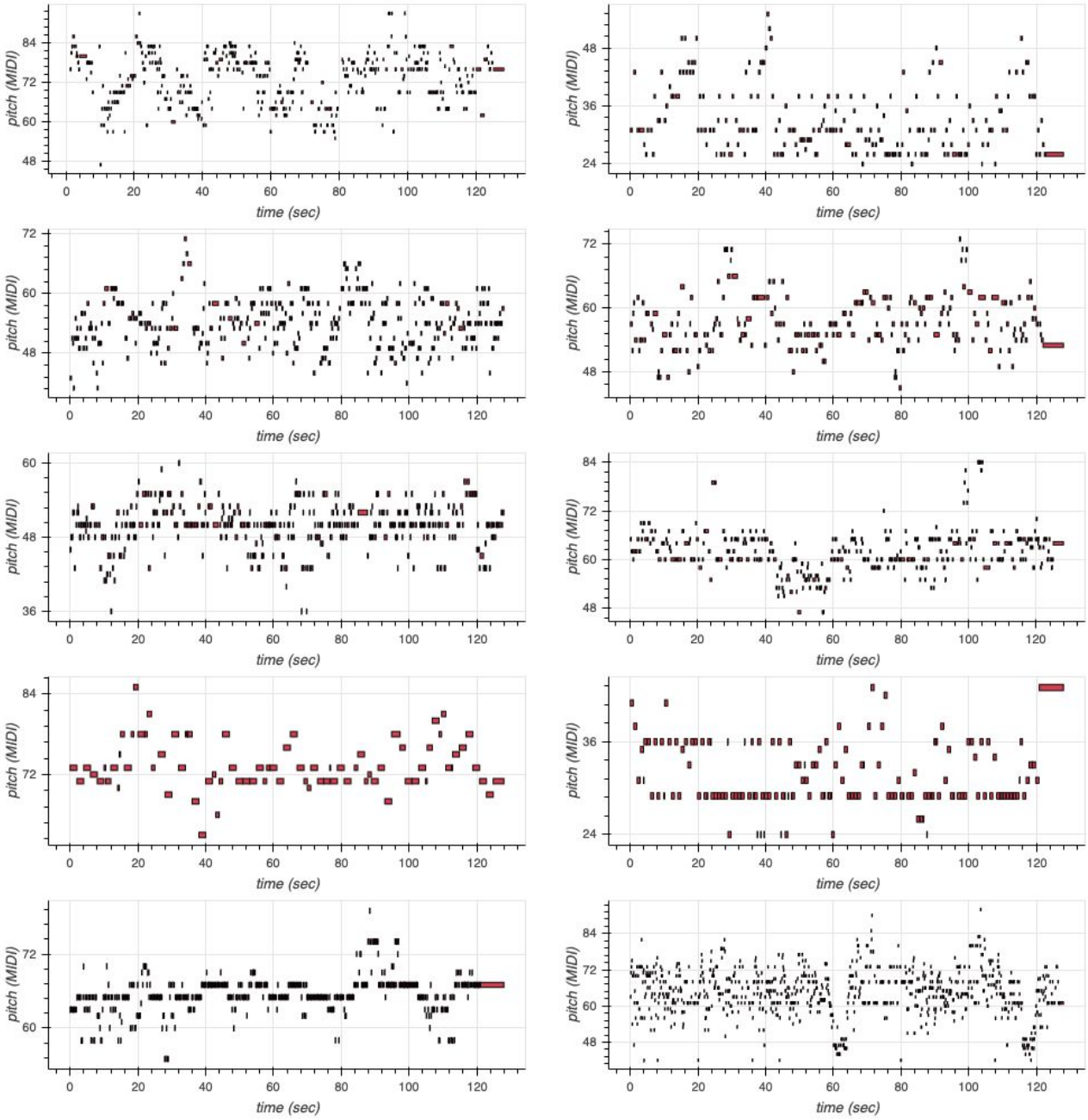

Figure 7. Additional piano rolls generated unconditionally by our diffusion model.

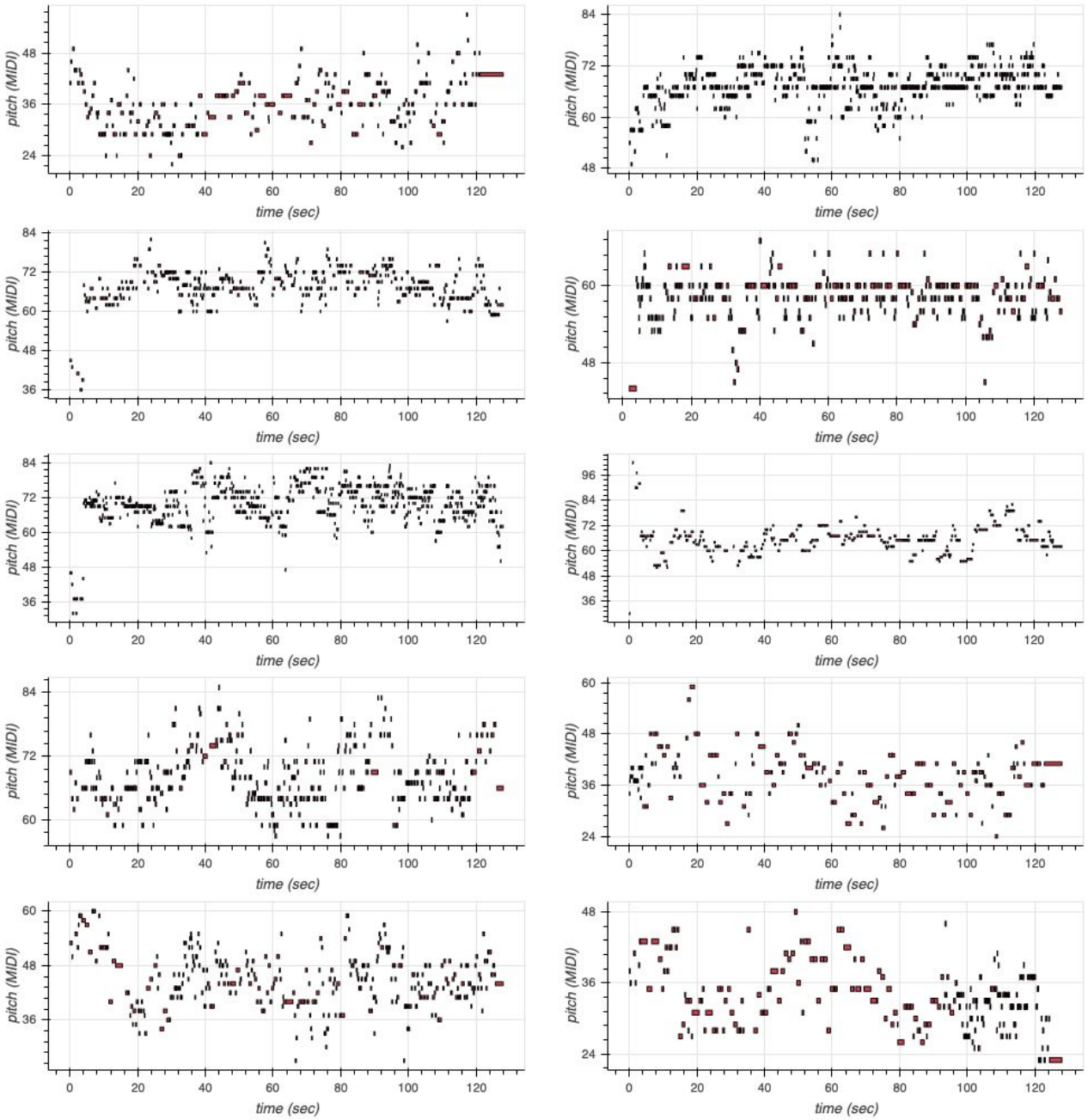

Figure 8. Additional piano rolls generated unconditionally by TransformerMDN.

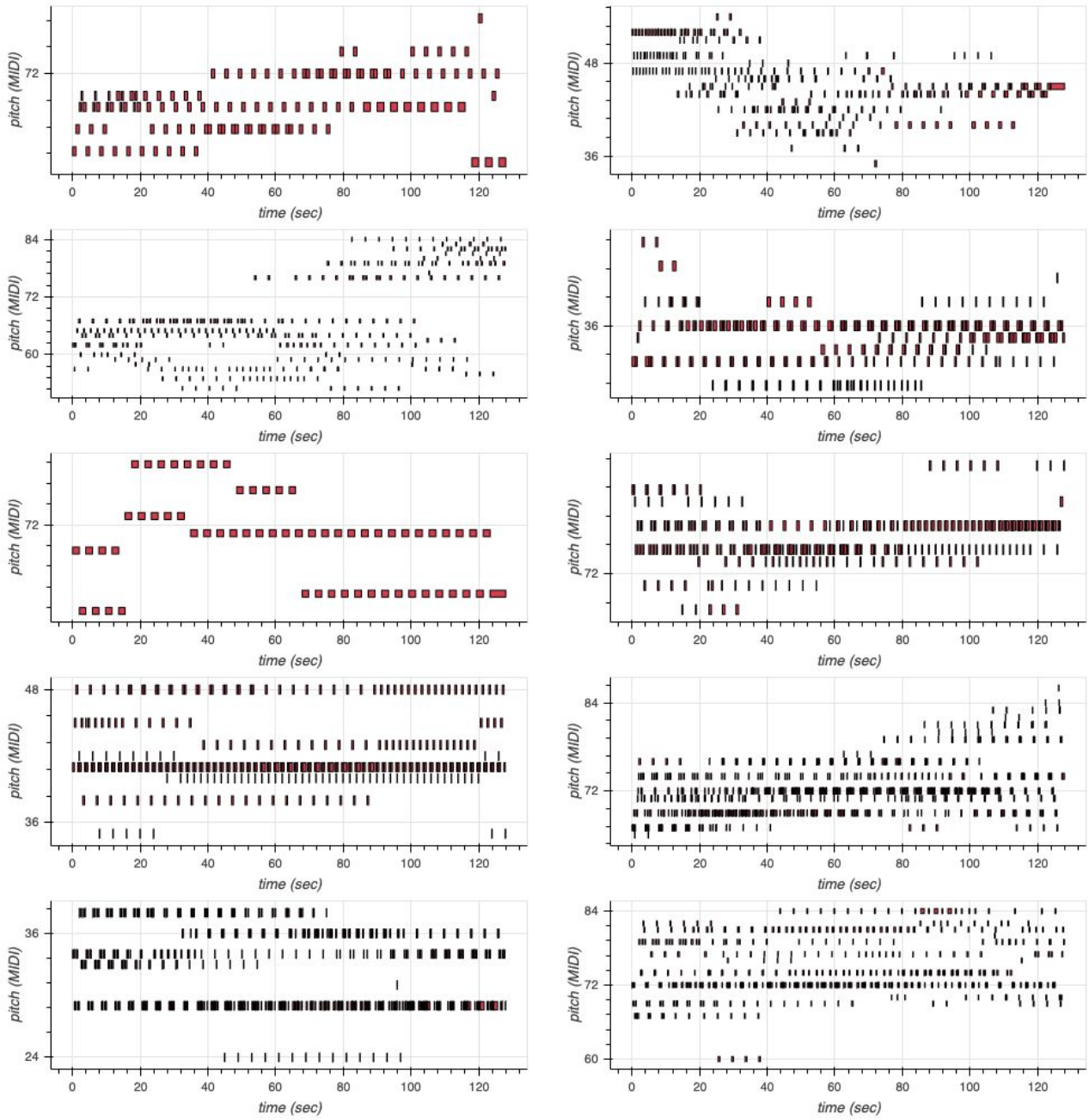

Figure 9. Additional piano rolls generated by performing spherical interpolation [30] between the first and last latent embeddings of sequences drawn from the test set.

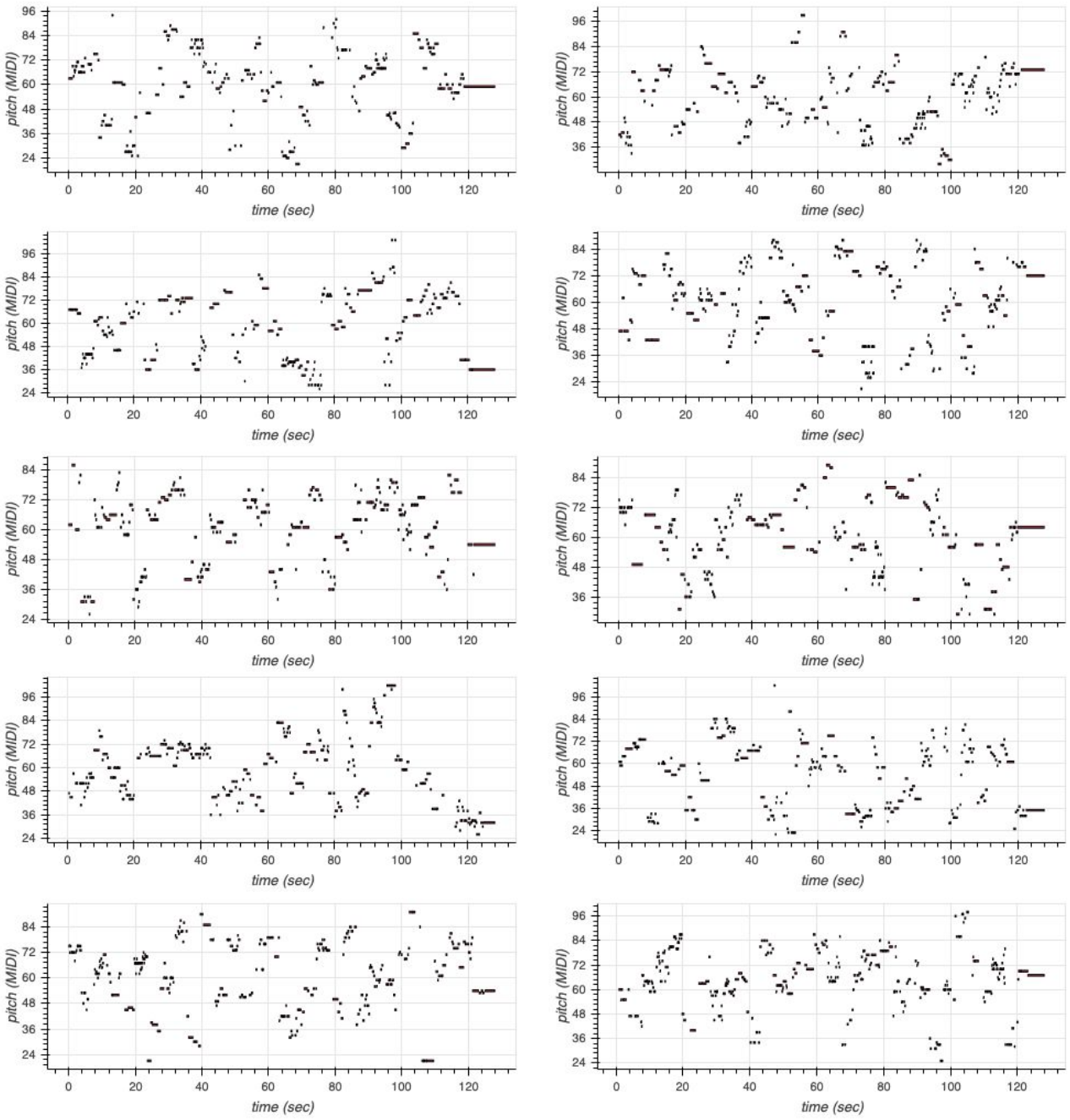

Figure 10. Additional piano rolls generated by sampling each latent embedding independently from the $\mathcal{N}(0, I)$ MusicVAE prior.

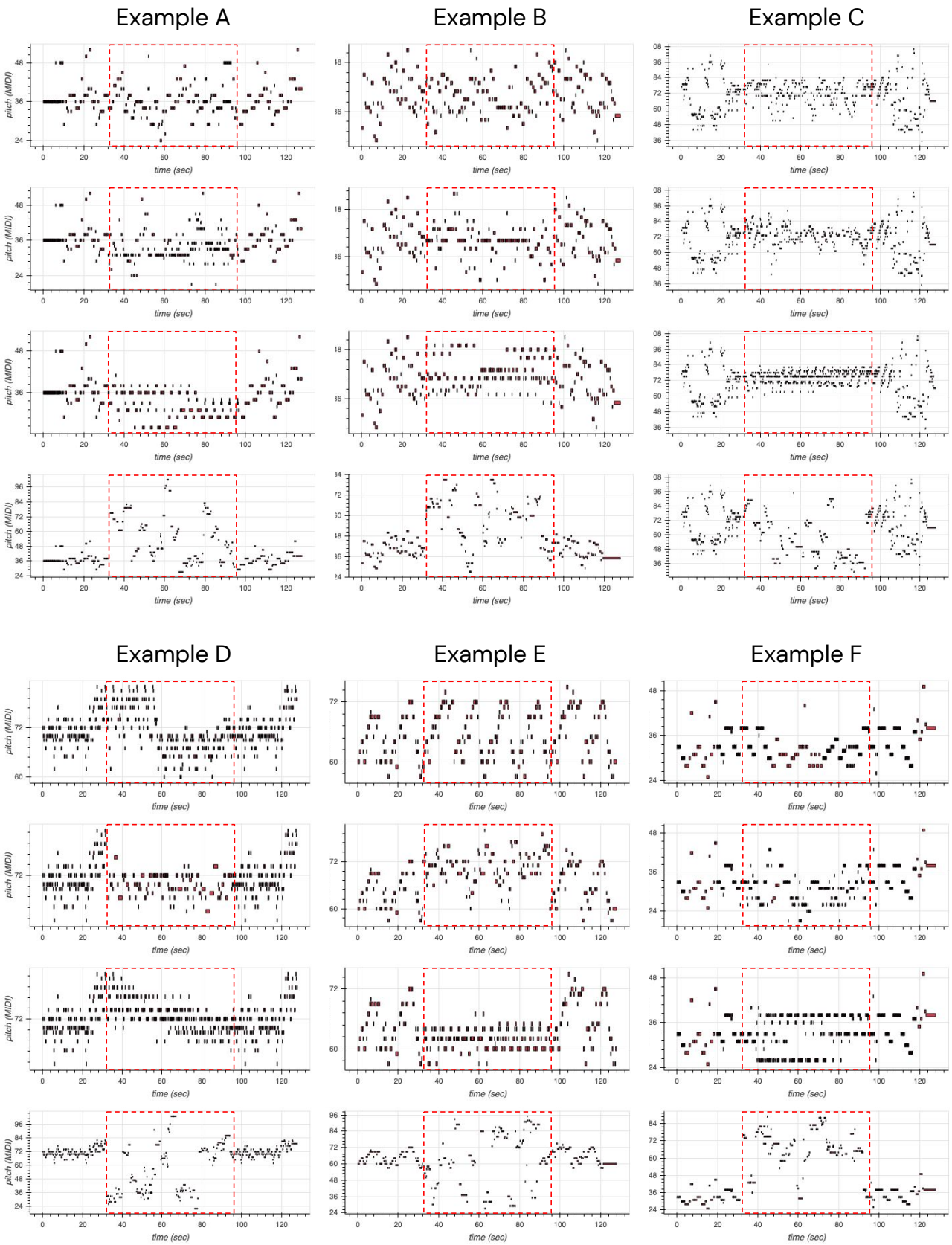

Figure 11. Additional piano rolls of infilling experiments. The first and last 256 melody tokens are held constant and the interior 512 tokens are filled in by the model (dashed red box). Original sample (first row), diffusion model (second row), interpolation (third row), sampling independently from the MusicVAE prior (fourth row).