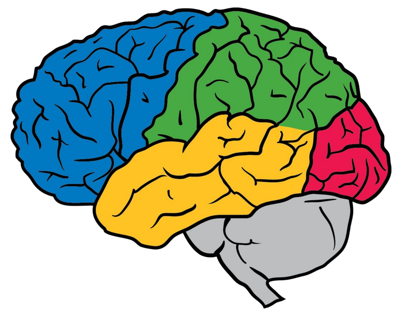


Audio Texture Synthesis with Random Neural Networks: Improving Diversity and Quality

Joseph M. Antognini*, Matthew Hoffman, Ron Weiss

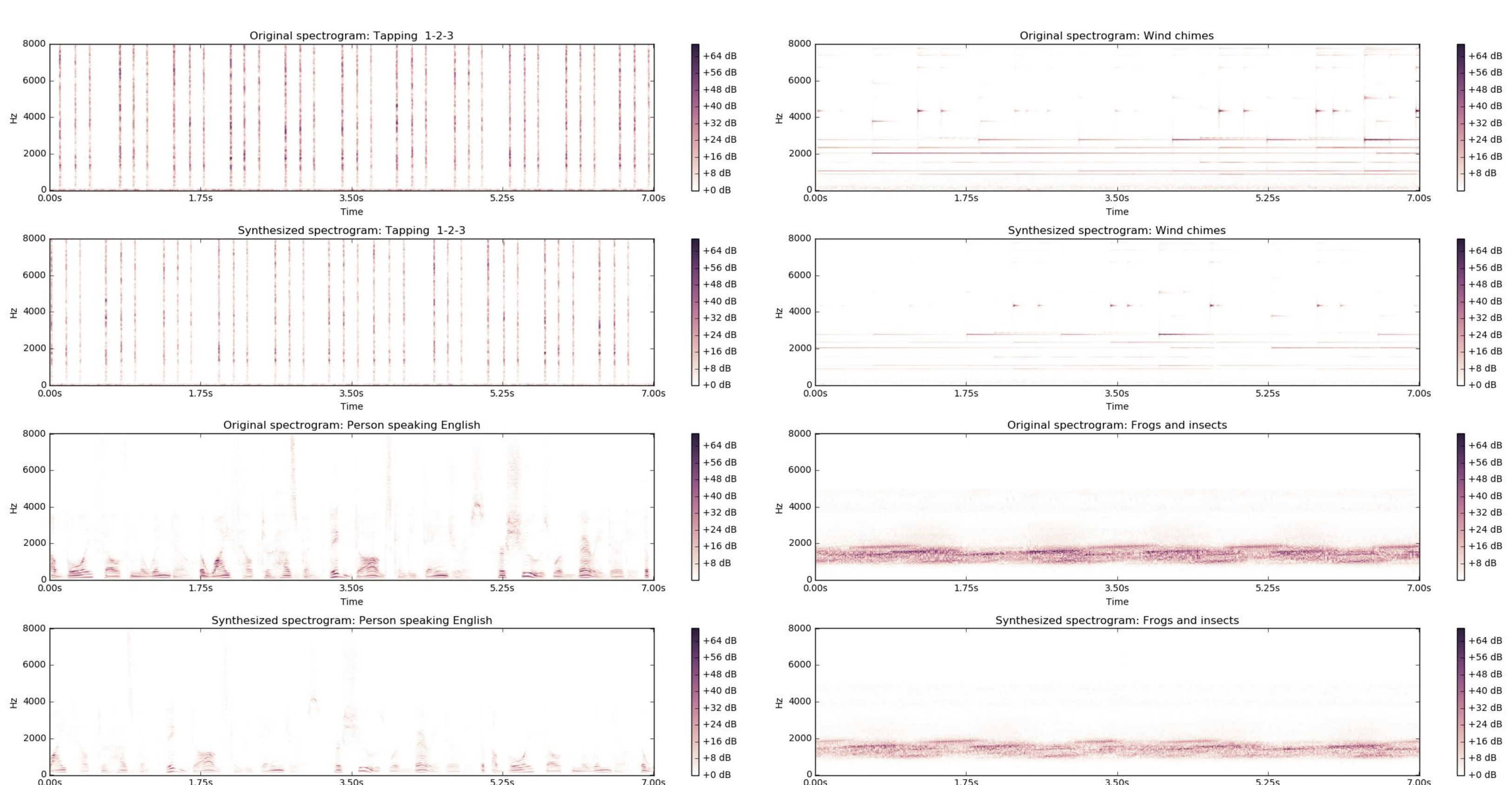
*Work done as part of Google AI Residency, g.co/airesidency

Summary

- We extend the technique of neural texture synthesis to the audio domain.
- We use an architecture consisting of six single-layer convolutional networks with random weights. The kernel widths of each layer are equally spaced logarithmically.
- We find that three loss terms are necessary for diverse, high quality textures:
  - A Gramian term to match the feature statistics of the target texture.
  - An autocorrelation term to reproduce rhythm.
  - And a diversity term that discourages exact feature matching.
- We synthesize spectrograms with L-BFGS optimization and then invert the spectrogram with Griffin-Lim.
- We analyze the quality of the resulting textures both quantitatively and qualitatively to show these three terms in the loss function are necessary to produce diverse, high quality textures.

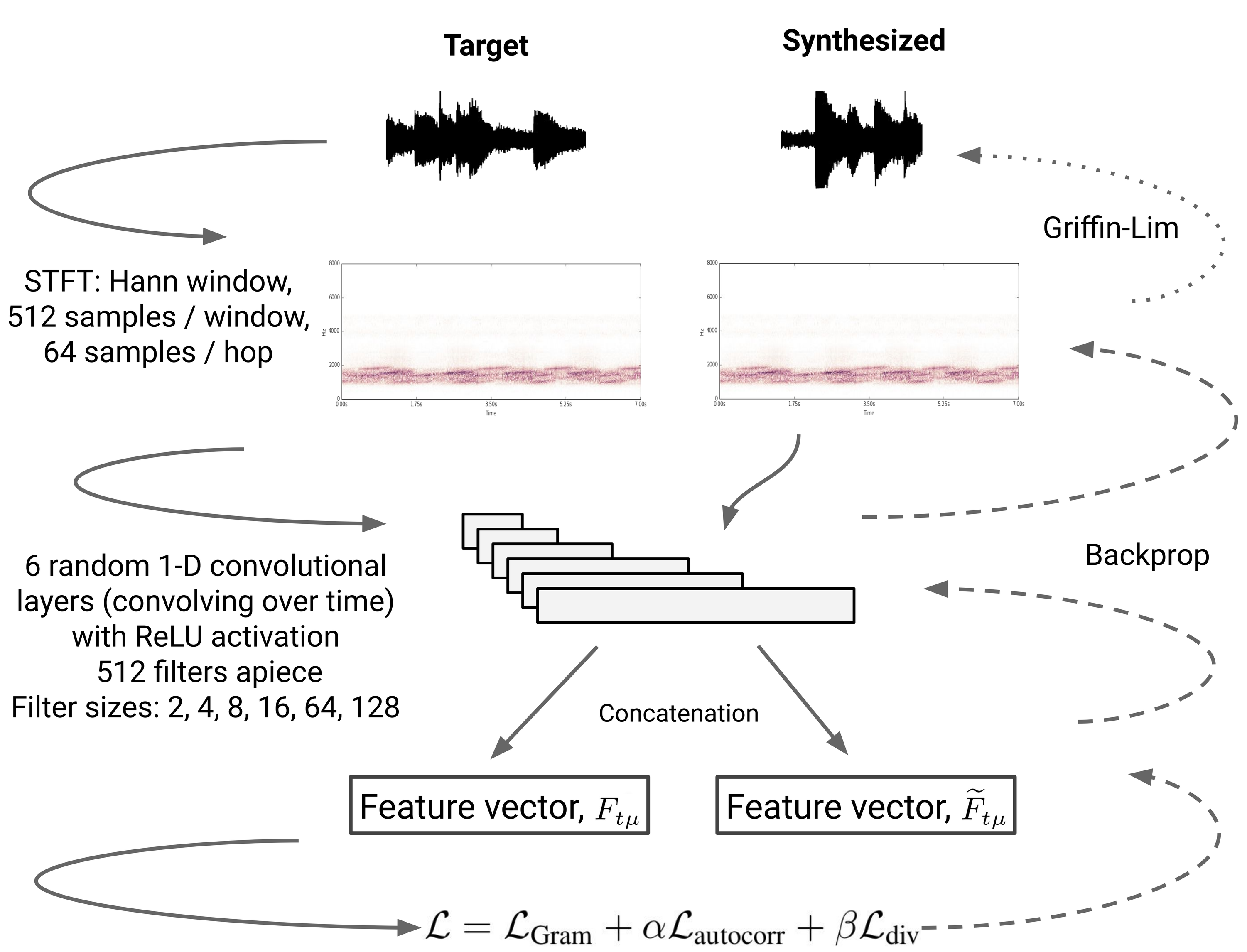
Problem & sample results

- *The problem of texture synthesis is to take a sample of some textured data (typically an image) and generate synthesized data which are perceptually similar, but are not identical to the original sample (or are not identical after a trivial transformation such as translation).*
- Spectrograms of four complex textures synthesized with our algorithm:



Samples can be heard at **https://antognini-google.github.io/audio_textures**

Signal processing & architecture



Loss terms

- *Gram loss:* Matches the Gram matrix of the feature activations of each convolutional layer. (Originally proposed by Gatys et al., 2015)

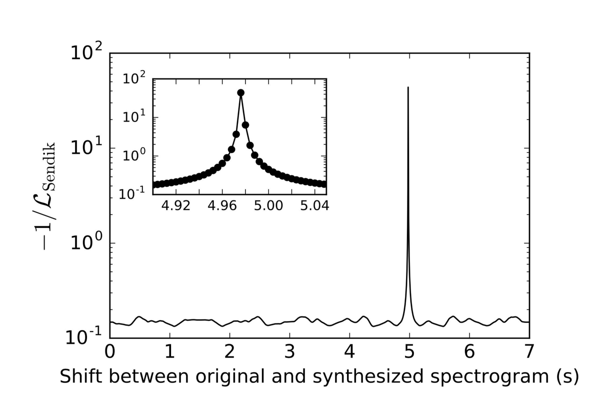
$$\mathcal{L}_{\text{Gram}} = \frac{\sum_{k,\mu,\nu}\left(G^k_{\mu\nu} - \widetilde{G}^k_{\mu\nu}\right)^2}{\sum_{k,\mu,\nu}\left(\widetilde{G}^k_{\mu\nu}\right)^2} \qquad G^k_{\mu\nu} = \frac{1}{T}\sum_t F^k_{t\mu}F^k_{t\nu}$$

- *Autocorrelation loss:* Matches the autocorrelation function of the target texture. We find that this term is necessary to synthesize rhythmic textures. (Originally proposed by Sendik & Cohen-Or, 2017.)

$$\mathcal{L}_{\text{autocorr}} = \frac{\sum_{k,\tau,\mu}\left(A^k_{\tau\mu} - \widetilde{A}^k_{\tau\mu}\right)^2}{\sum_{k,\tau,\mu}\left(\widetilde{A}^k_{\tau\mu}\right)^2} \qquad A^k_{\tau\mu} = \mathscr{F}^{-1}_f\left[\mathscr{F}_t[F^k_{t\mu}]\,\mathscr{F}_t[F^k_{t\mu}]^*\right]$$

- *Shift-invariant diversity loss:* Penalizes the optimizer for synthesizing textures which are identical to the original but shifted in time. Without this term the optimizer can reproduce a shifted version of the original texture.

$$\mathcal{L}_{\text{div}} = \max_s\left(\frac{\sum_{k,t,\mu}\left(\widetilde{F}^k_{t\mu}\right)^2}{\sum_{k,t,\mu}\left(F^k_{t+s,\mu} - \widetilde{F}^k_{t\mu}\right)^2}\right)$$



Optimization

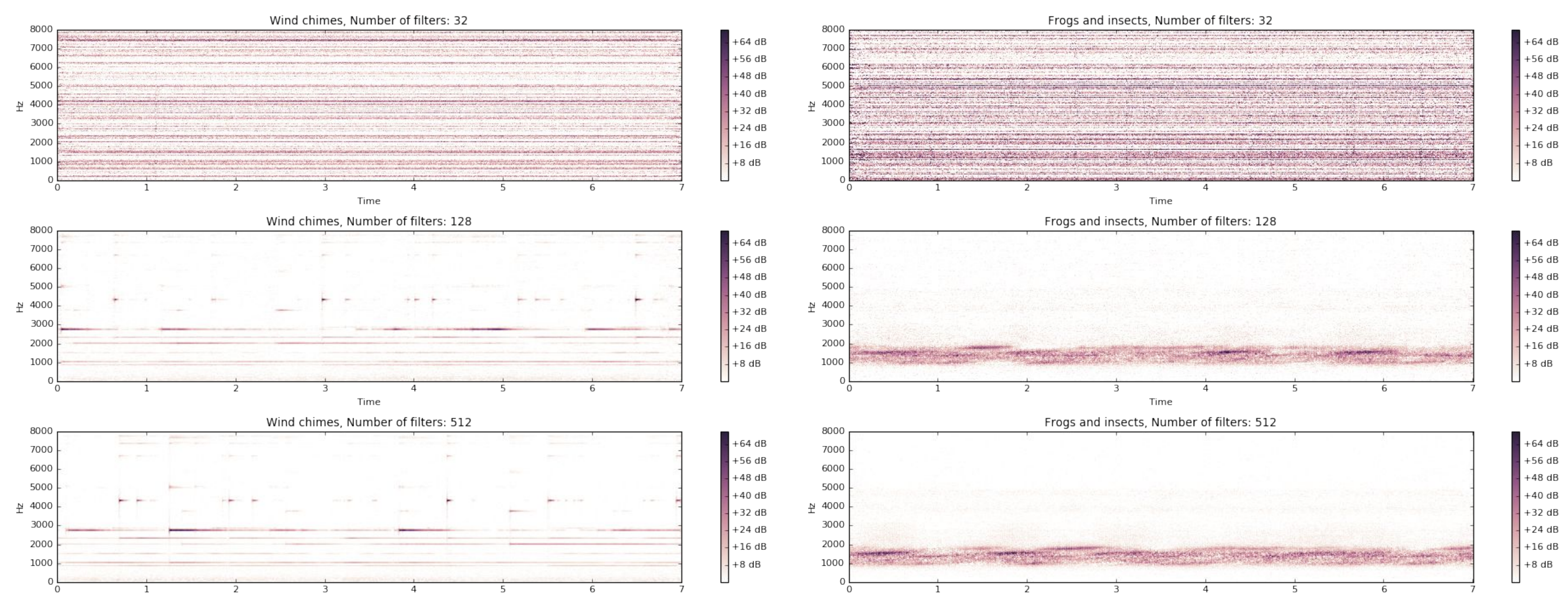
- We backprop through to the spectrogram and optimize the spectrogram with 2000 iterations of the L-BFGS algorithm. We then invert the spectrogram with 500 iterations of the Griffin-Lim algorithm to obtain audio.

References

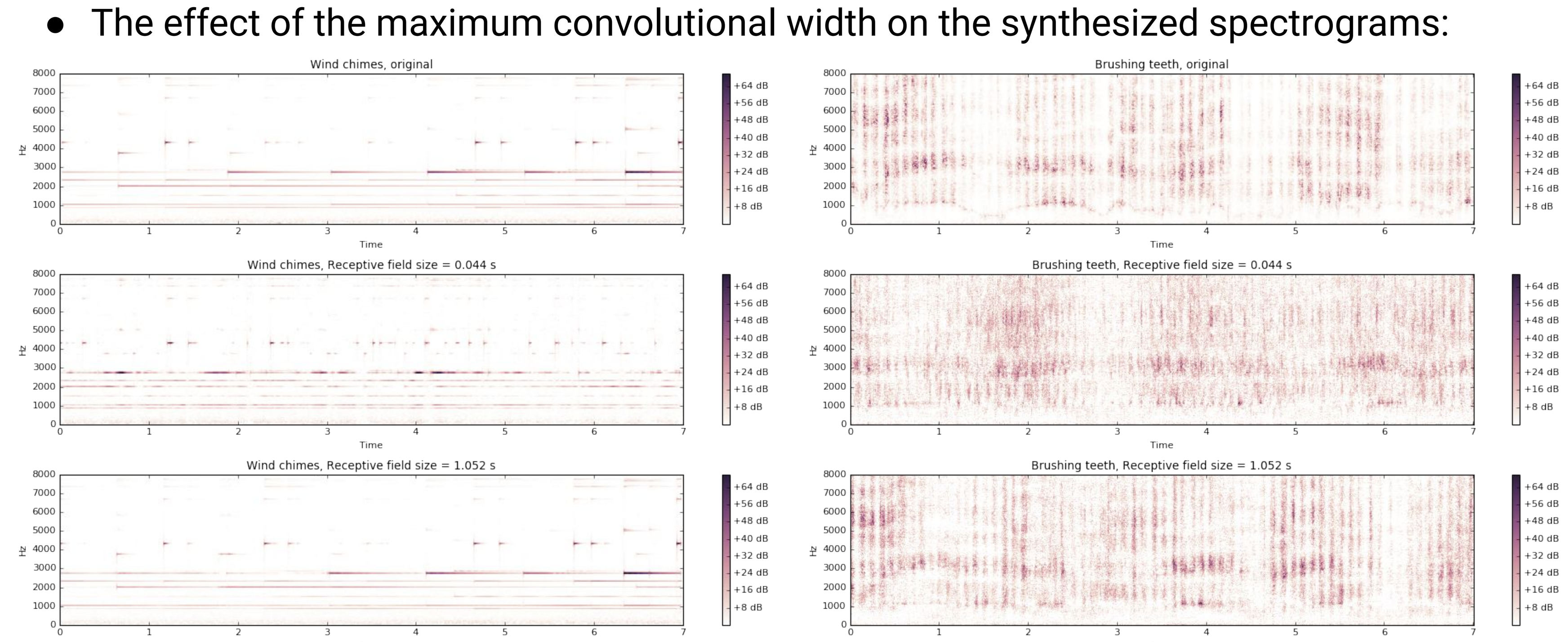
Griffin, D., & Lim, J., "Signal estimation from modified short-time fourier transform", 1984
Gatys, L., Ecker, A., & Bethge, M., "Texture synthesis using convolutional neural networks", 2015
McDermott, J. & Simoncelli, E., "Sound texture perception via statistics of the auditory periphery", 2011
Portilla, J. & Simoncelli, E., "A parametric texture model based on joint statistics of complex wavelet coefficients", 2000
Sendik & Cohen-Or, "Deep correlations for texture synthesis", 2017
Ulyanov, D., & Lebedev, V., "Audio texture synthesis and style transfer", 2016

Qualitative evaluation

- The effect of the number of filters on the synthesized spectrograms:



- The effect of the maximum convolutional width on the synthesized spectrograms:



Quantitative evaluation

- We introduce a "VGGish score" in analogy to the "Inception score" common in the GAN literature.
  - The VGGish score attempts to measure both the diversity and the quality of the synthesized textures.
  - It measures the KL divergence between the output distribution of the VGGish neural network on a set of real textures and synthesized textures:

$$\mathcal{S}_{\text{VGGish}} \equiv \exp\left[\mathbb{E}_x\left[\text{KL}\left(p_{\text{VGGish}}(y|\widetilde{x})\,||\,p_{\text{VGGish}}(y|x)\right)\right]\right]$$

- We measure an "autocorrelation score" given by the squared difference of the autocorrelations of the spectrograms to determine how well rhythms are matched.
- We measure a "diversity score" given by the max of the inverse squared differences between the spectrograms for all shifts.
  - Unlike the loss terms, these scores are computed on the *spectrograms* rather than on the *features*.
- Using the Gram loss alone produces the highest VGGish scores, but the autocorrelation loss is necessary to achieve a good autocorrelation score. Adding in the diversity loss in turn increases the diversity and improves the VGGish score.

	VGGish ($\times 10^{-4}$)			Autocorrelation			Diversity		
	Rthm.	Ptch.	Other	Rthm.	Ptch.	Other	Rthm.	Ptch.	Other
Spectrograms recovered via Griffin-Lim	9.7	12.6	7.1	7.4	0.54	2.9	21.4	29.7	22.7
McDermott & Simoncelli (2011)	16.7	33.2	8.3	542.0	408.1	421.9	**1.6**	**1.6**	**2.0**
Ulyanov & Lebedev (2016)	13.4	26.8	10.0	40.6	23.3	27.4	2.9	3.0	3.3
$\mathcal{L}_{\text{Gram}}$	**9.9**	**16.8**	**7.3**	29.0	9.7	6.5	2.4	3.0	3.5
$\mathcal{L}_{\text{Gram}} + \mathcal{L}_{\text{autocorr}}$	17.8	21.3	17.9	13.3	7.4	15.6	3.4	5.4	5.0
$\mathcal{L}_{\text{Gram}} + \mathcal{L}_{\text{autocorr}} + \mathcal{L}_{\text{div}}$ ($\beta = 10^{-5}$)	14.5	23.0	12.2	13.0	**2.3**	7.2	3.8	6.8	4.4
$\mathcal{L}_{\text{Gram}} + \mathcal{L}_{\text{autocorr}} + \mathcal{L}_{\text{div}}$ ($\beta = 10^{-3}$)	14.9	19.0	10.0	**4.7**	3.7	**7.1**	5.0	4.9	3.9

The diversity-quality tradeoff

- The diversity of the synthesized textures can be controlled by the weight on the diversity loss term.
- As diversity of the synthesized textures increases (due to a higher weight on the diversity term), the quality as measured by the VGGish score decreases. (Higher VGGish score is lower quality.)