SeamlessGAN: Self-Supervised Synthesis of Tileable Texture Maps

IEEE Transactions on Visualization and Computer Graphics (TVCG), 10.1109/TVCG.2022.3143615

Carlos Rodríguez-Pardo[1,2], Elena Garcés[1,3]

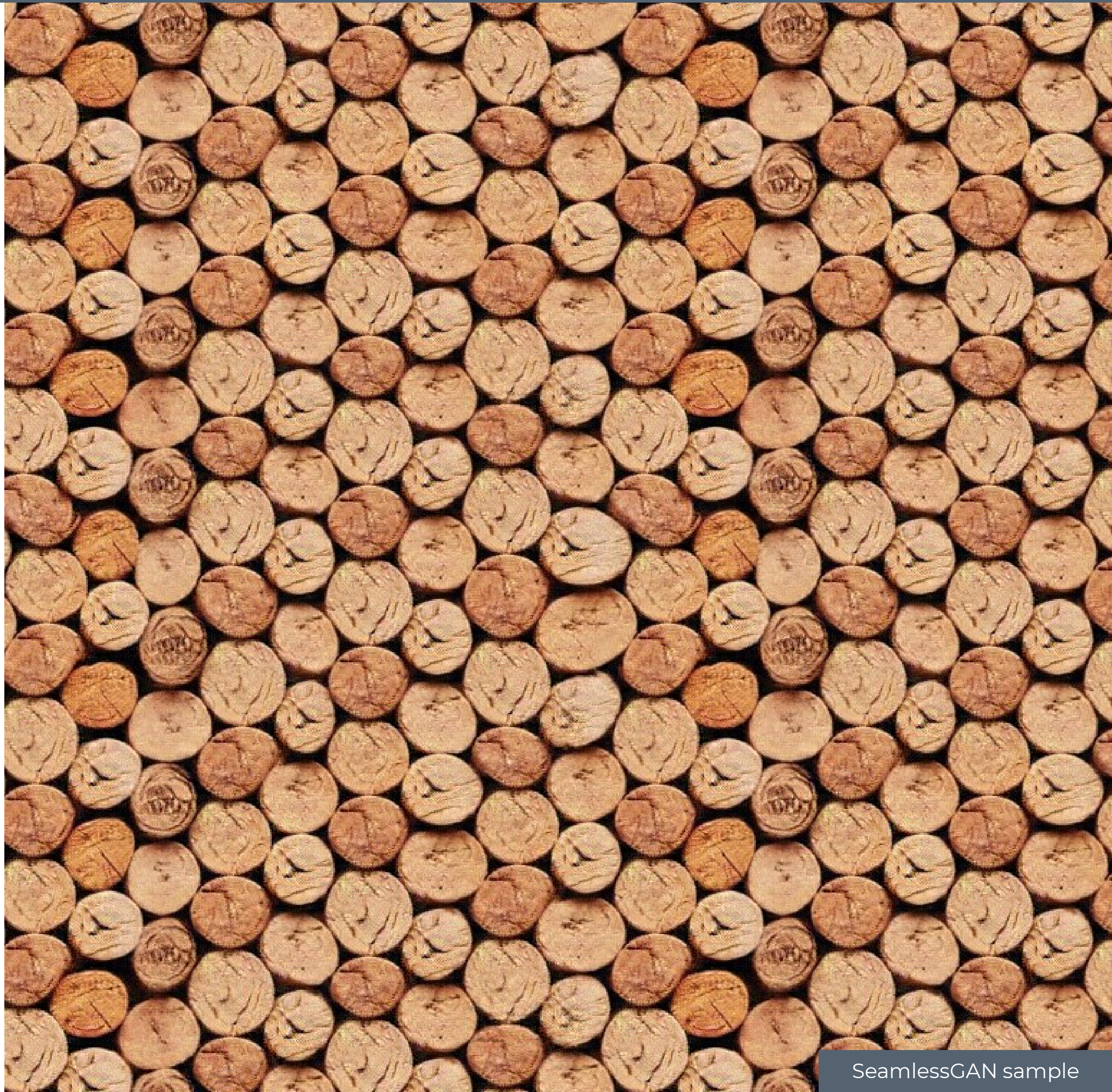
SEDDI[1]

uc3m | Universidad Carlos III de Madrid[2]

Universidad Rey Juan Carlos[3]

Outline

- **Motivation**
- **Related Work**
  - Texture Synthesis
  - Single-Image GAN
- **SeamlessGAN**
  - Training
  - Sampling
- **Evaluation**
- **Results**
- **Conclusions & Future Work**

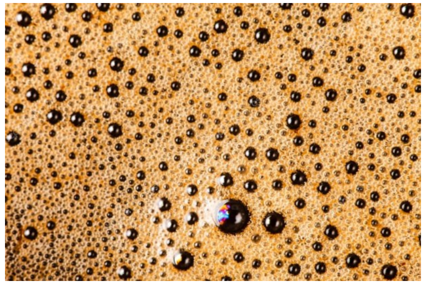


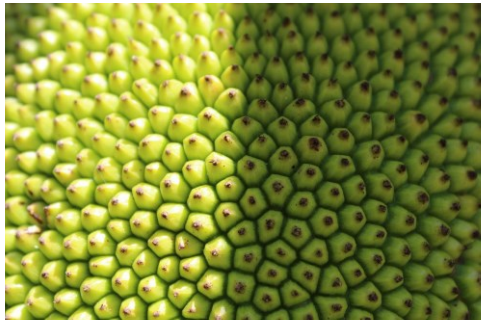
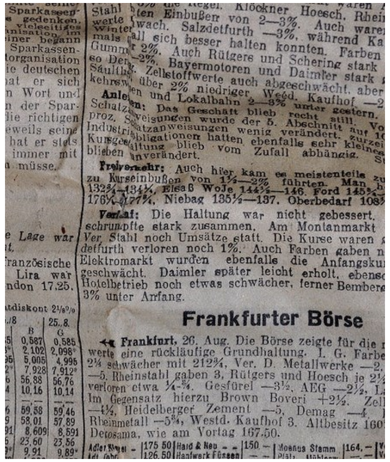
Motivation

In order to create realistic 3D environments, we need accurate digital materials.

Motivation: Textures

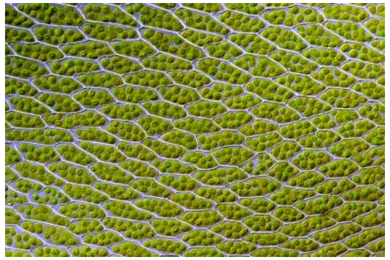
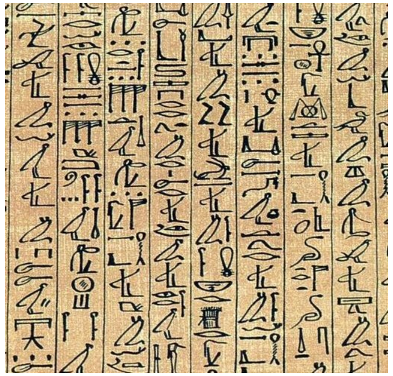












Textures are **everywhere** in the real world, from man-made environments to natural settings.

Motivation: Digital Materials

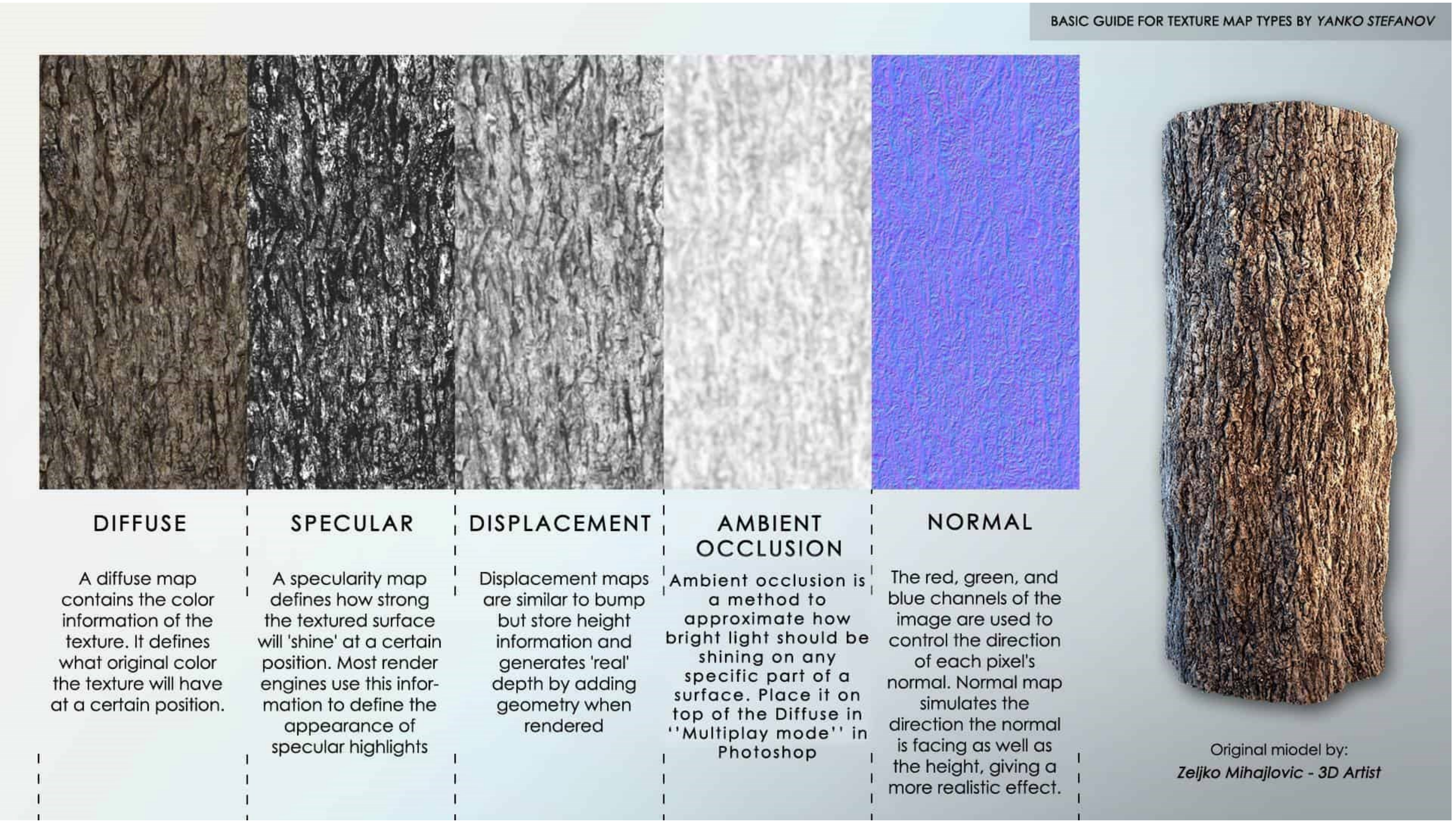
CAD Software

Videogames and movies

Virtual Reality Applications

Automatic texture generation algorithms are essential for digital environment creation.

Motivation: SVBRDFs



Digital textures can be modelled through Spatially-Varying BRDFs (or ***texture stacks***), with multiple maps.
Each map represents a distinct property of the material.
Multiple *texture maps* are needed for enhancing the realism of digital materials.

Motivation: SVBRDFs

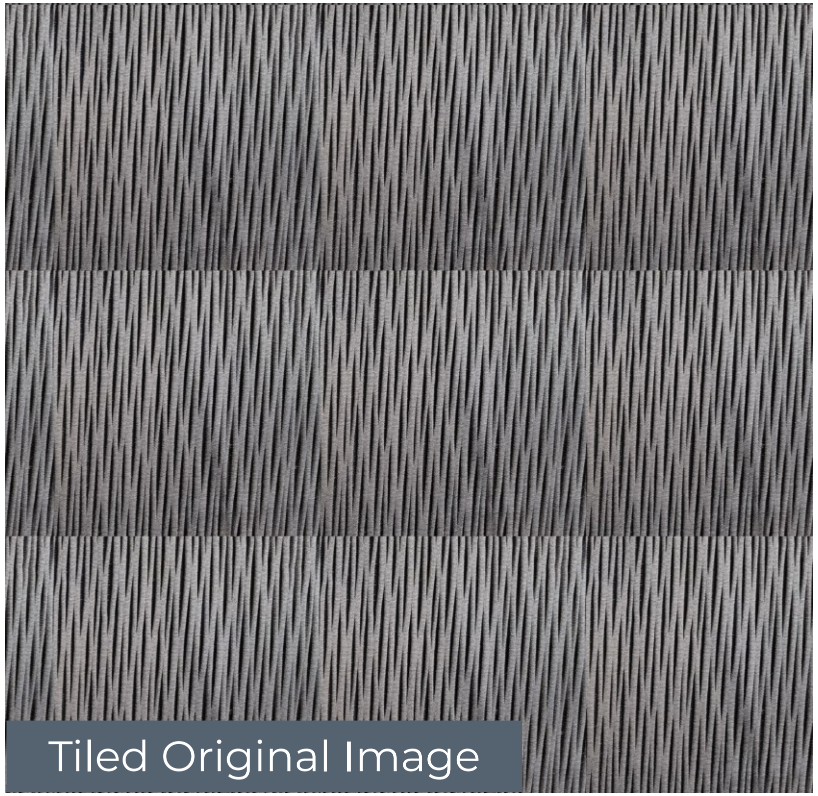
Using only a Base Color Map

Using Multiple Maps

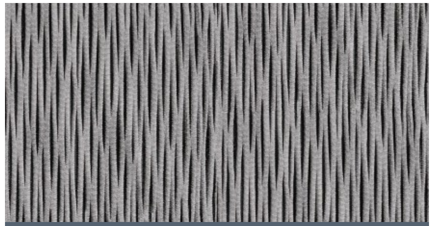
Multiple maps increase the realism of digital materials.
Renders and materials obtained using SEDDI Textura.

Motivation: Tileable Textures

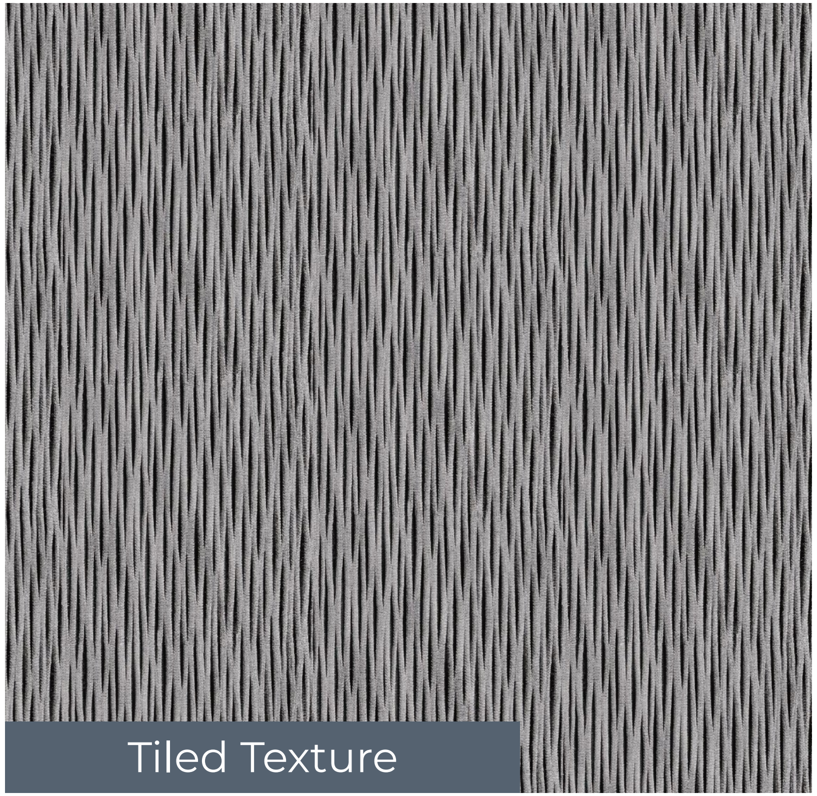
Original Image



Tiled Original Image



Tileable Texture



Tiled Texture

Textures in digital environments must be ***tileable*** in order to provide realism.
They also need to be small for memory efficiency.
Turning an image into a tileable texture is a very challenging task.

Motivation: Tileable Textures

Original Image

Tileable Texture Stack

Renders and materials obtained using *SEDDI Textura*.

Tileable texture stacks allow for realistic material representations.

How can we generate a tileable texture stack from a single sample?

Related Work

- **Texture Synthesis**
  - Parametric
  - Non-Parametric
- **Tileable Texture Synthesis**

⚠ Lots of work published on this topic, **not an exhaustive list**

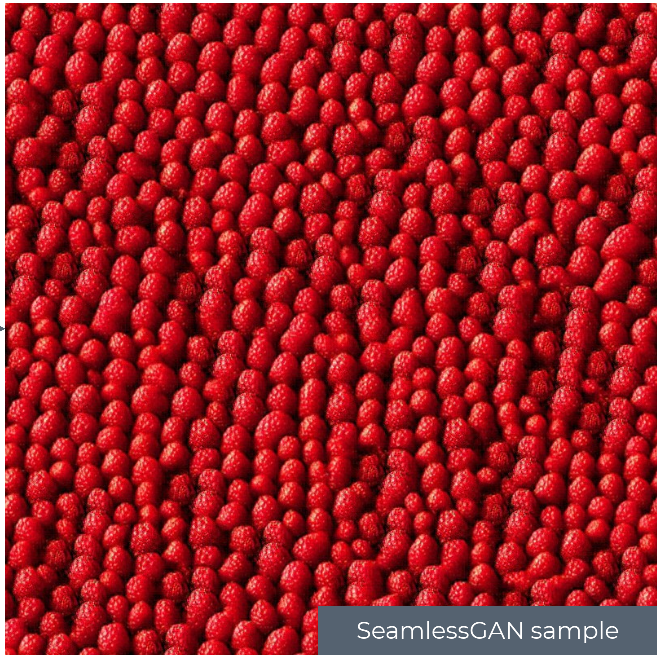
SeamlessGAN sample

Related Work: Texture Synthesis

Input Image

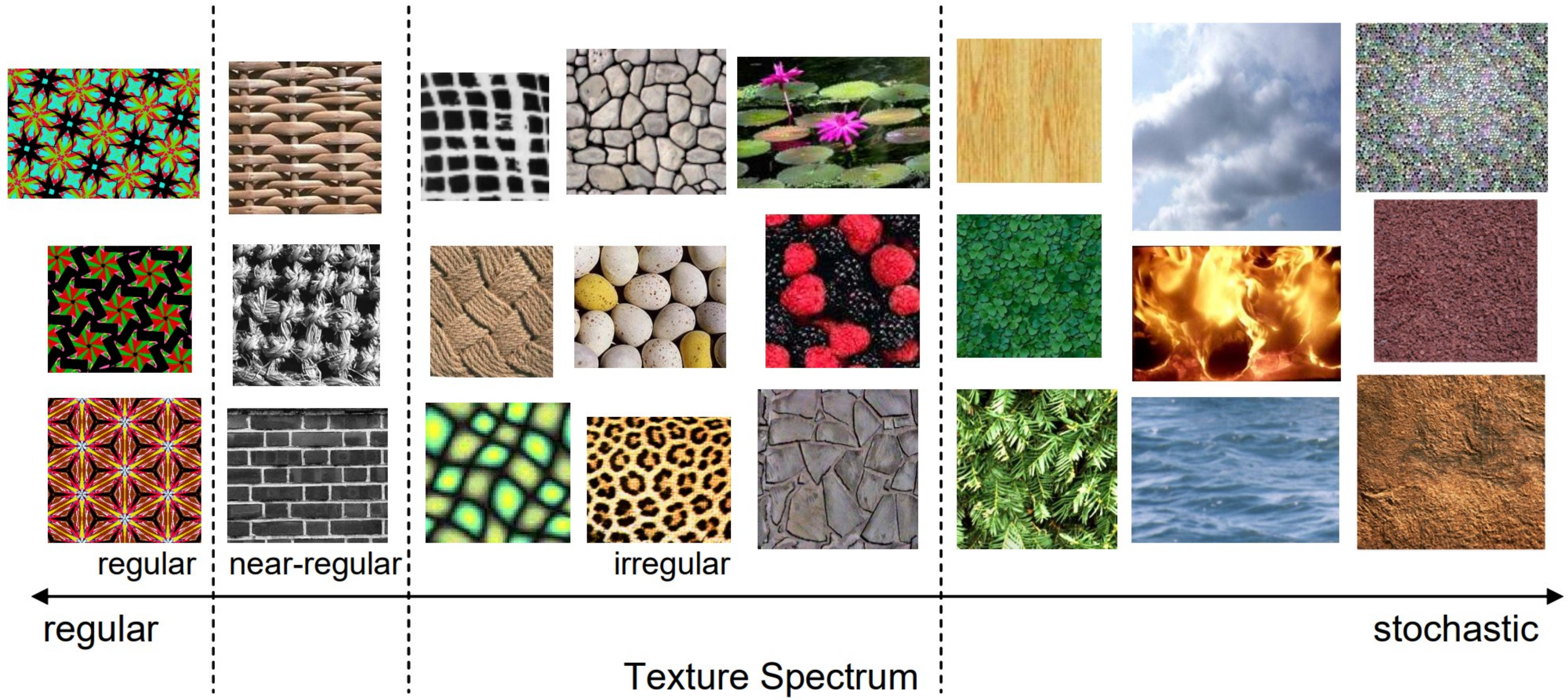
Synthesis Algorithm

Synthesized Texture



Texture synthesis is a special case of image synthesis where the images are characterized by **repeating patterns**. This puts special requirements on the synthesis algorithms.

Related Work: Texture Synthesis



Textures lie on a spectrum of regularity.

Each method works best for textures that lie on specific parts of the spectrum.

Generality is hard to achieve.

Figure from: **A Comparison Study of Four Texture Synthesis Algorithms on Near-regular Textures** [Lin∗ et al., 2004]

Related Work: Texture Synthesis

Non-Parametric Texture Synthesis

Output image contains *patches* that are present in the input image.

Approaches:
- Image quilting
- GraphCuts
- Genetic Algorithms
- Optimization
- Patchmatch

Typically:
🏃 Faster
👎 Less expresive

Parametric Texture Synthesis

A model learns statistics from the input image, and generates an output that matches those statistics.

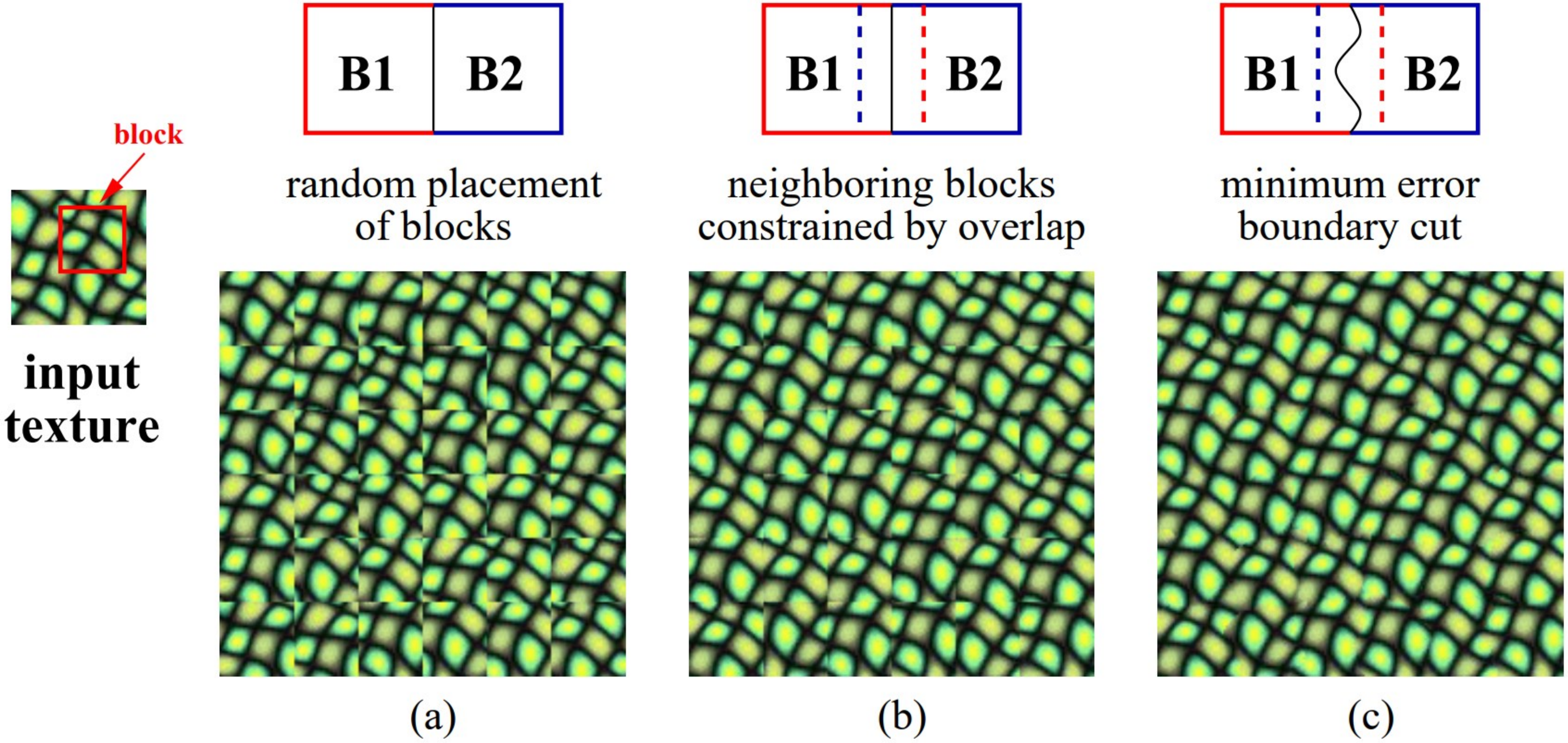
Approaches:
- Gradient Descent Optimization
- Activations on pre-trained CNNs
- **GANs**

Typically:
🧠 Rely on learning
👀 More flexible and better results
🐢 More expensive and slow

GANs are SOTA for image synthesis, also for textures.

Related Work: Non-Parametric Texture Synthesis

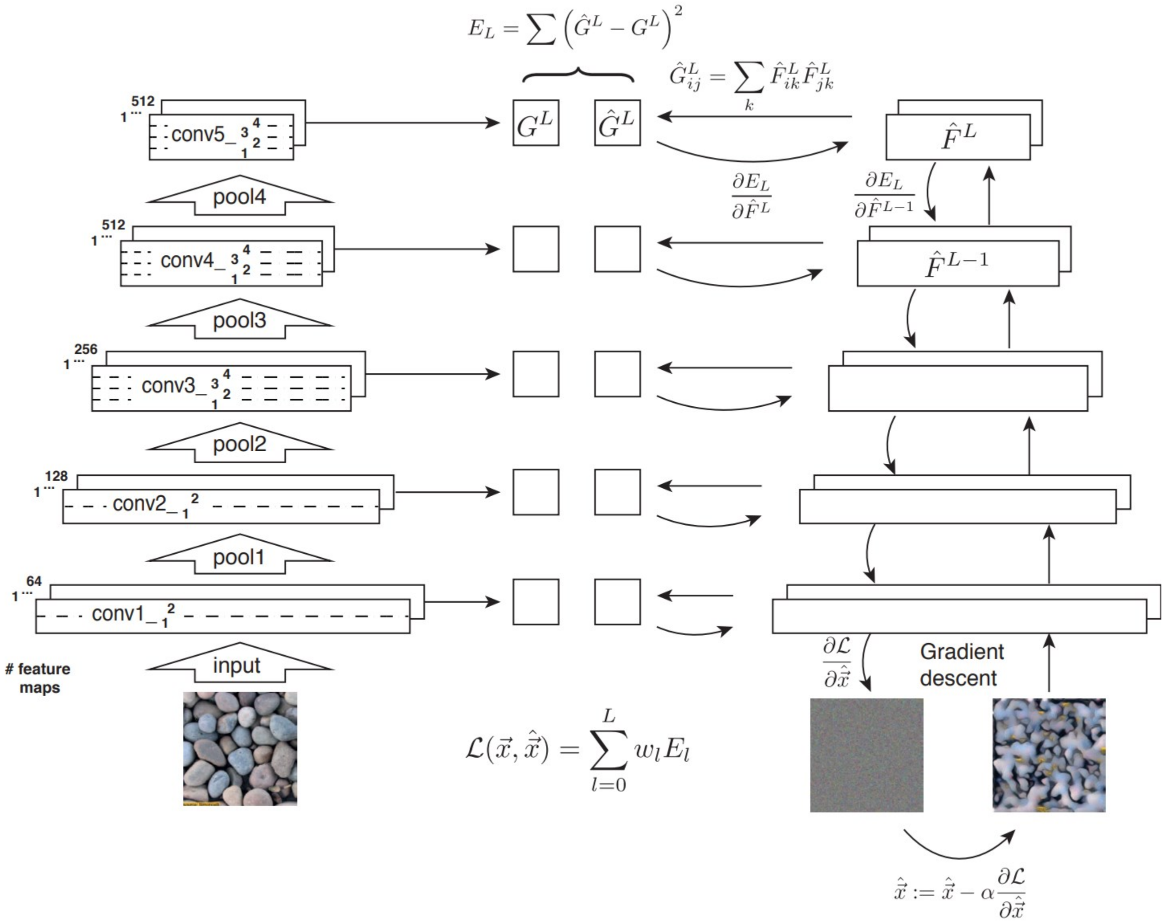
Image Quilting for Texture Synthesis and Transfer [Efros and Freeman, 2001]



Seminal work on non-parametric texture synthesis.

Related Work: Parametric Texture Synthesis

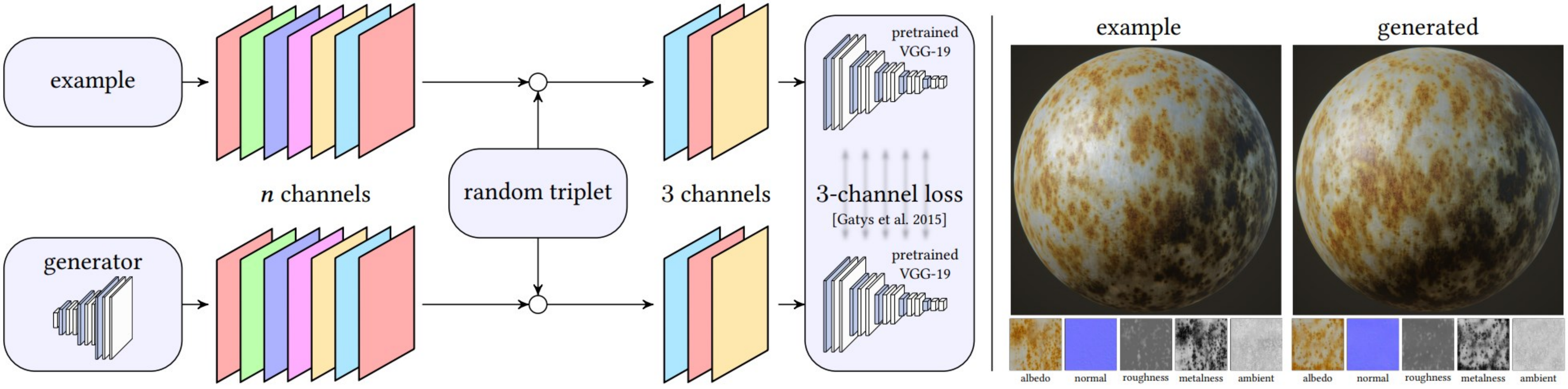
Texture synthesis using convolutional neural networks [Gatys et al., 2015]



Pre-trained CNNs provide a strong prior for texture synthesis.

Related Work: Multi-Layer Texture Synthesis

Passing Multi-Channel Material Textures to a 3-Channel Loss [Chambon et al., 2021]



Very recent extension to CNN-based optimization methods **for multi-map synthesis.**
Limited by the expressivity of a VGG network as an image descriptor.

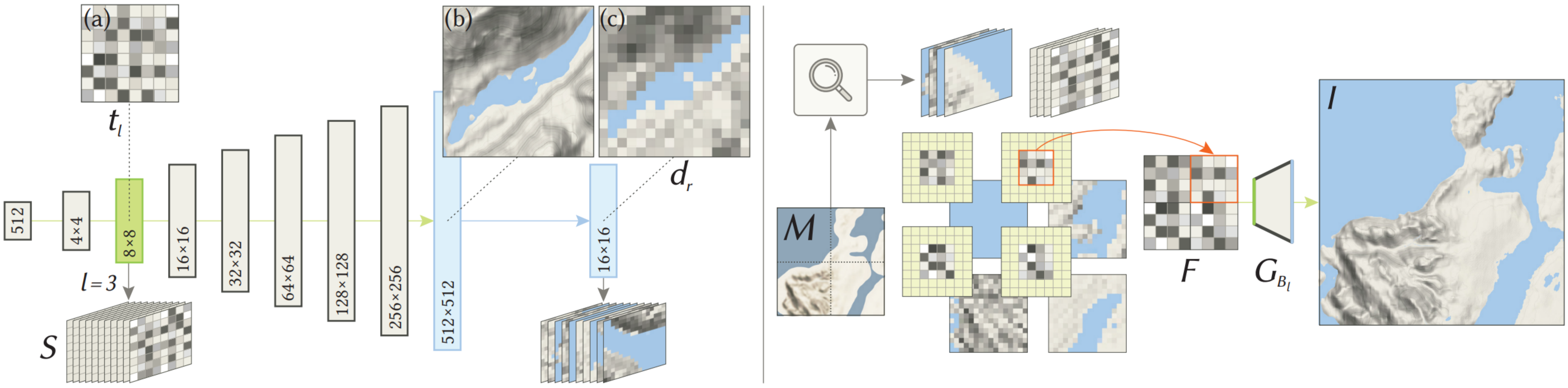
Related Work: Parametric Texture Synthesis

⚠ **Non-stationary texture synthesis by adversarial expansion** [Zhou et al., 2017]

Single-Image GANs for texture synthesis. Powerful training process which generates SOTA, high-resolution results. 🐢 Several hours of training per texture.

Related Work: Parametric Texture Synthesis

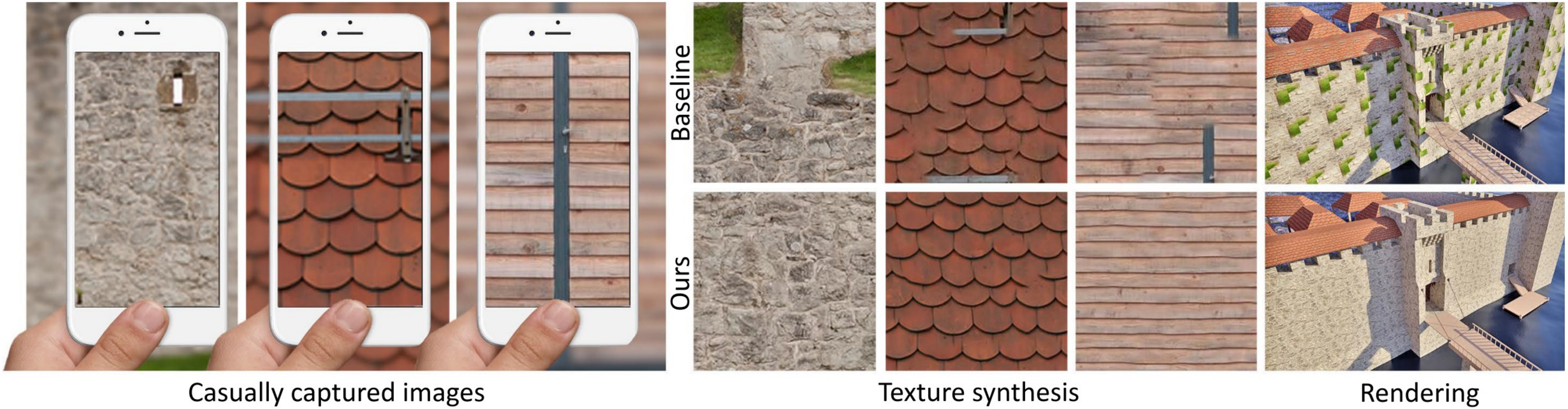
⚠ **TileGAN: Synthesis of Large-Scale Non-Homogeneous Textures** [Frühstück et al., 2019]



Latent space manipulation for texture blending. Trained on a large dataset of satellite images.

Related Work: Non-Parametric Tileable Texture Synthesis

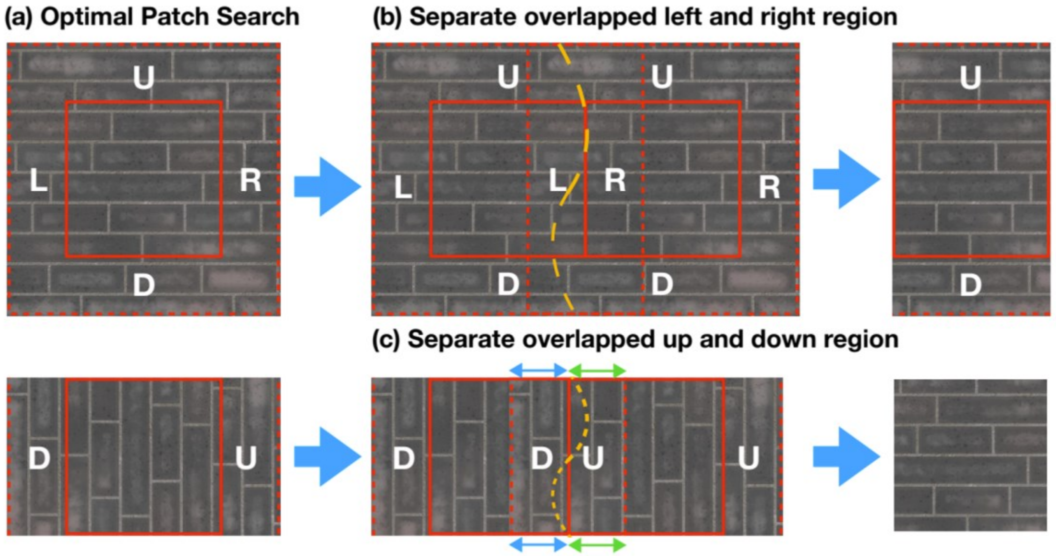
Texture Stationarization: Turning Photos into Tileable Textures [Moritz et al., 2017]



Seamless synthesis by PatchMatch.

Related Work: Non-Parametric Tileable Texture Synthesis

Inverse rendering for complex indoor scenes: Shape, spatially-varying lighting and svbrdf from a single image [Li et al., 2020]



Our graph-cut based method for tileable texture synthesis. We first stitch the left-right boundaries of textures and then the up-down boundaries. When stitching up-down boundaries, we add a hard constraint so that left-right boundaries remain tileable.

Seamless synthesis by stitching patches using GraphCuts.

Related Work: Non-Parametric Tileable Texture Synthesis

Procedural Stochastic Textures by Tiling and Blending [Deliot and Heitz, 2019]

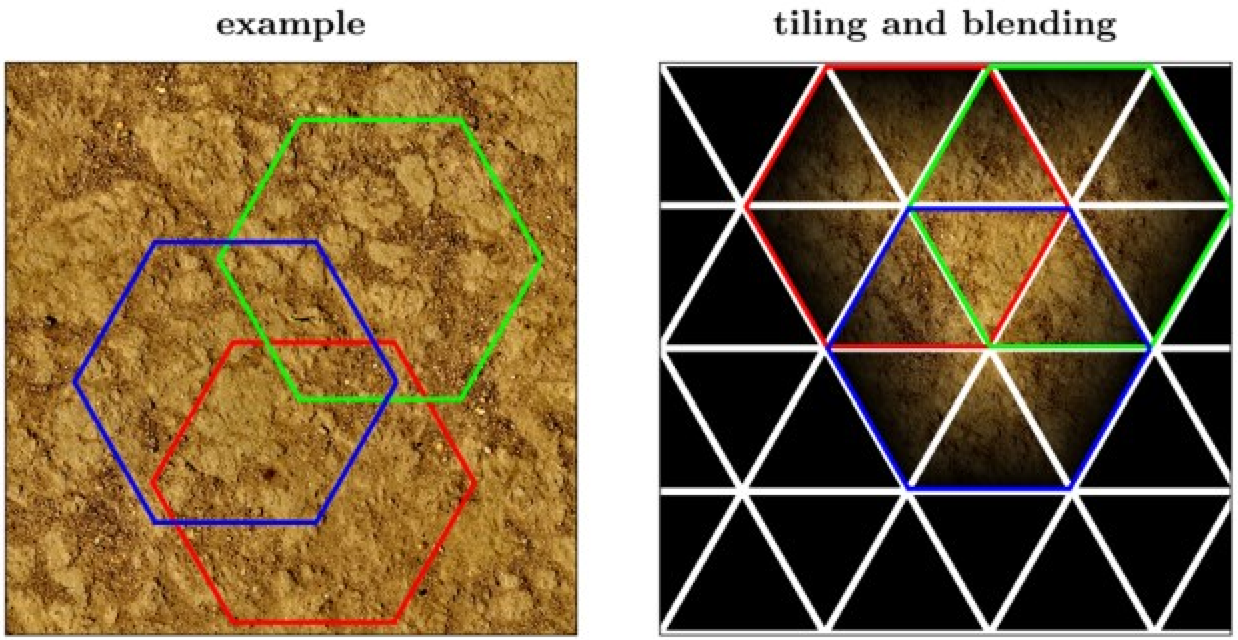
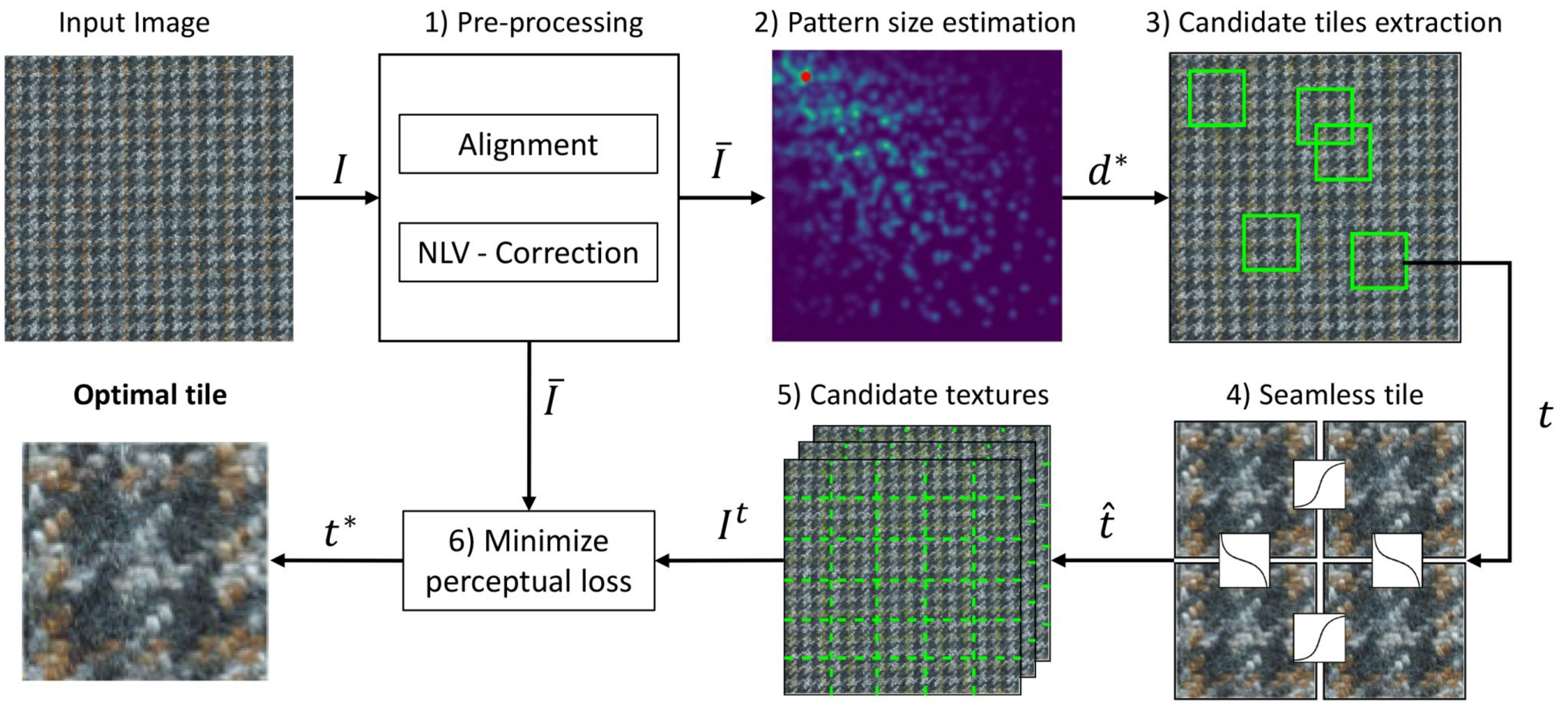


Figure 1.2. Tiling and blending. *Each pixel is obtained by blending three tiles from the example.*

Seamless synthesis by tiling and blending. Works in real time.
Only works for stochastic textures.

Related Work: Non-Parametric Tileable Texture Synthesis

Automatic Extraction and Synthesis of Regular Repeatable Patterns [Rodríguez-Pardo et al., 2019]

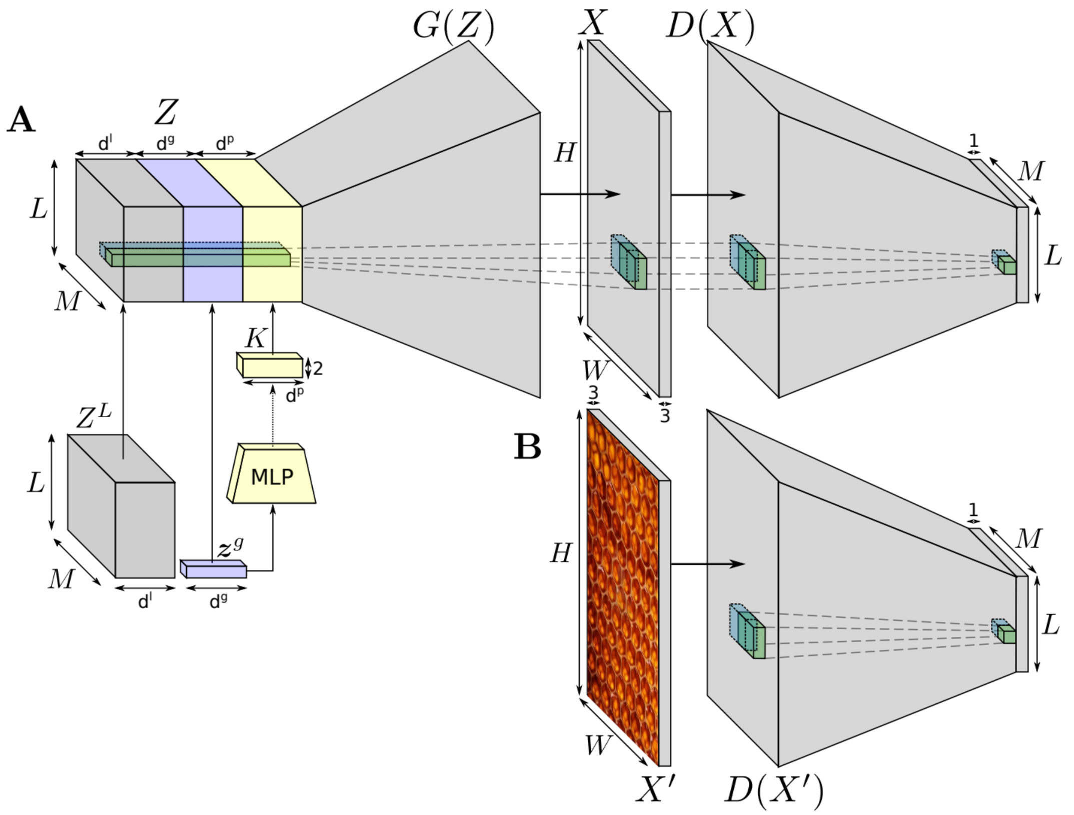


Optimal patch found using pre-trained CNNs. Tiling done by blending.

👎 Does not really work for non-regular textures.

Related Work: Parametric Tileable Texture Synthesis

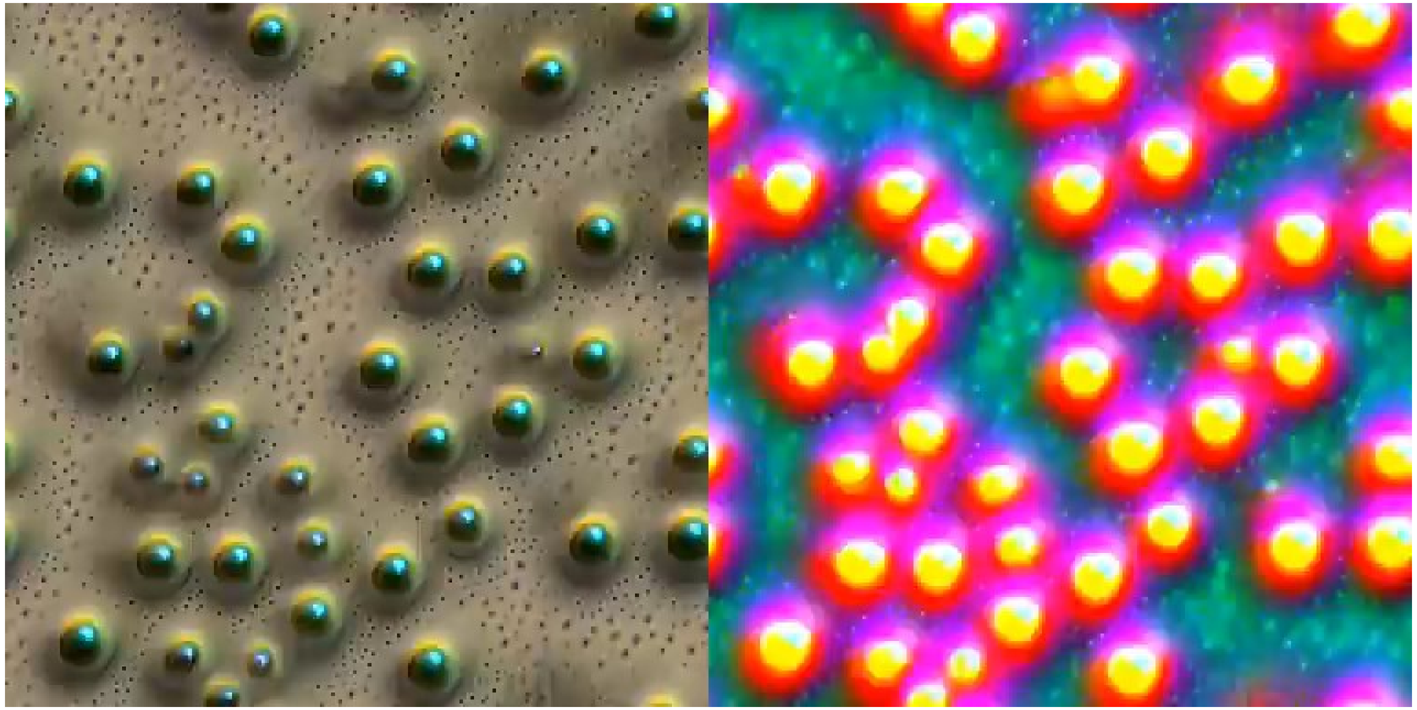
Learning Texture Manifolds with the Periodic Spatial GAN [Bergmann et al., 2017]



GANs for tileable texture synthesis. Can learn from 1 or more textures at a time.
Tileability enforced through periodicity.

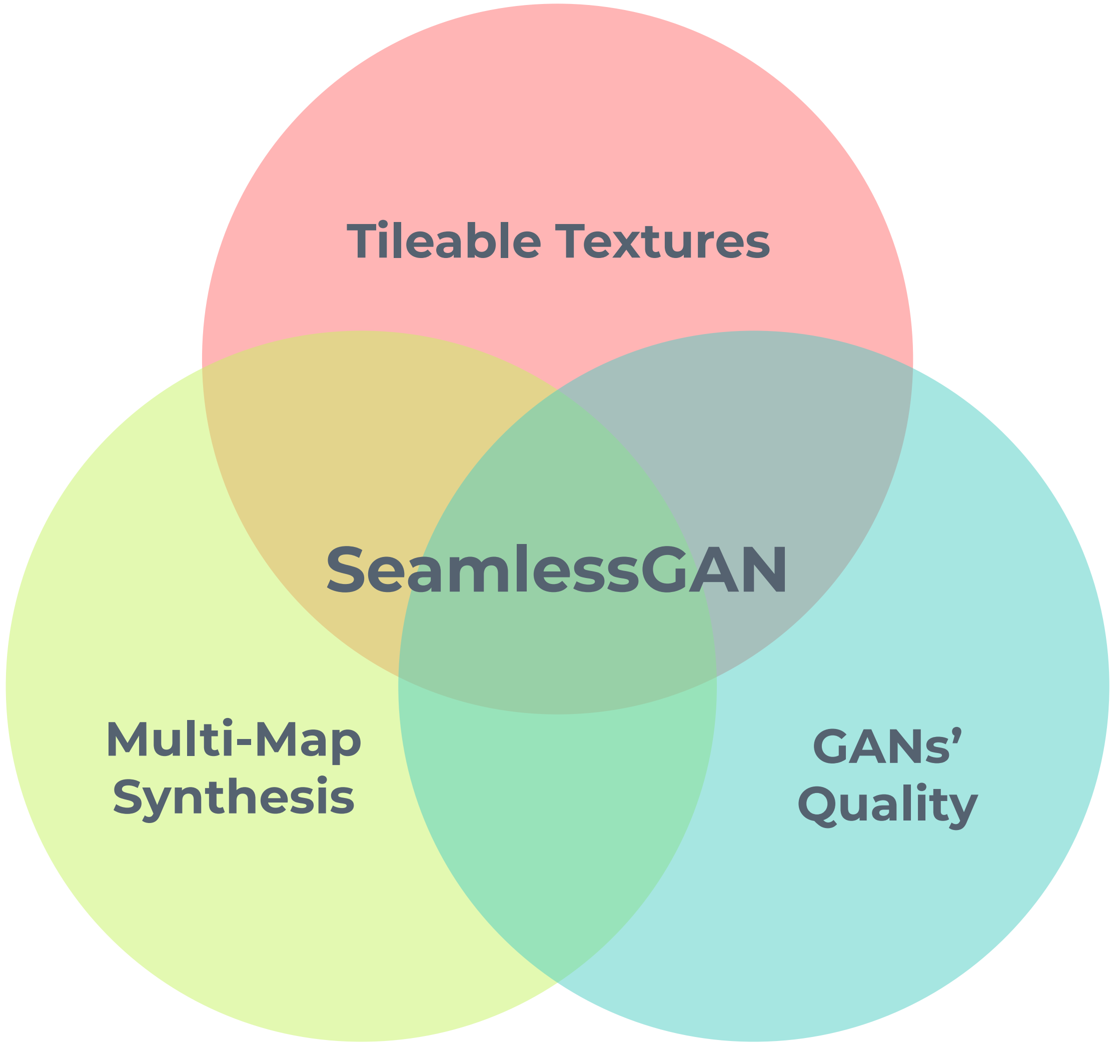
Related Work: Parametric Tileable Texture Synthesis

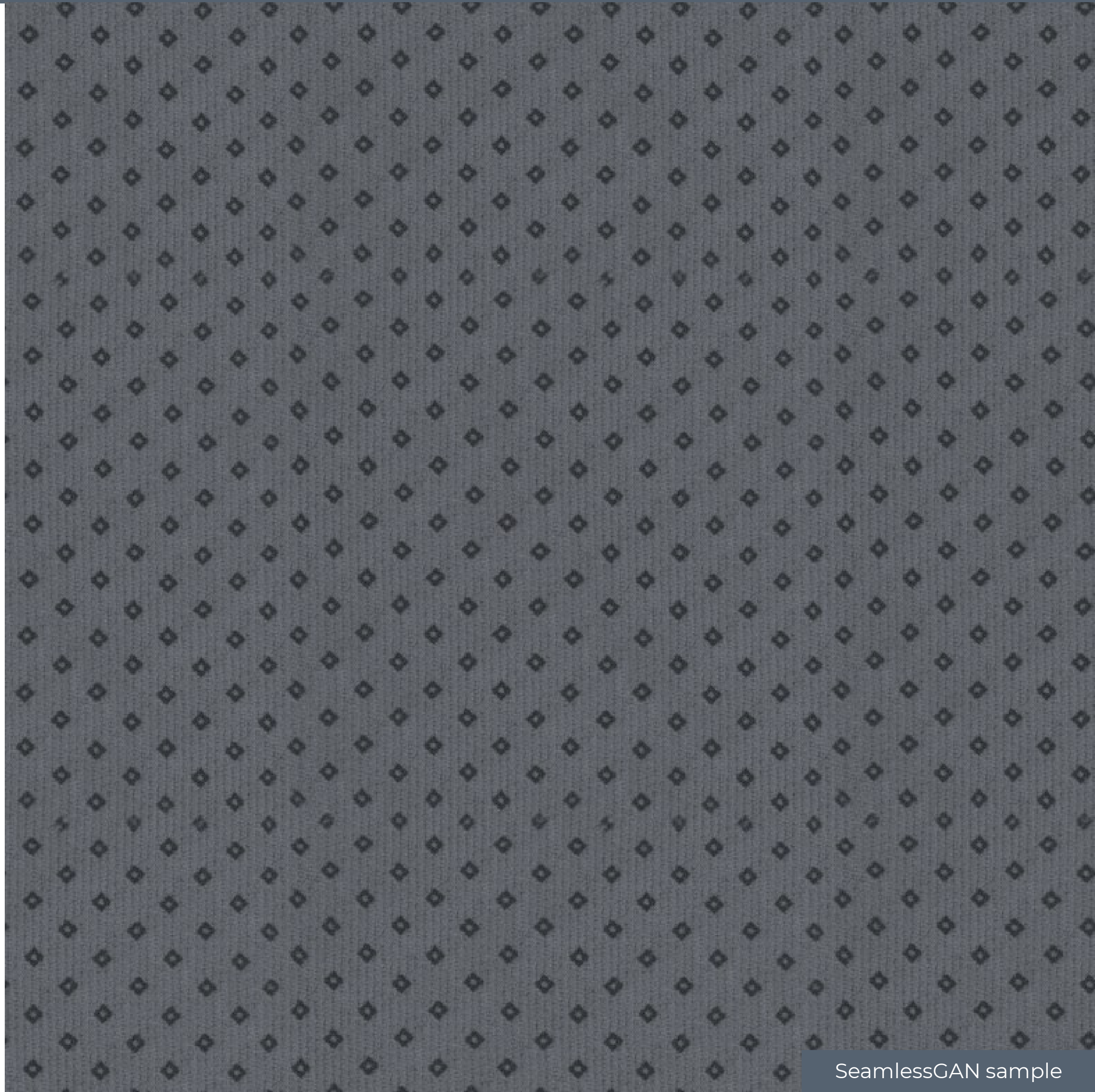
Self-Organising Textures [Niklasson et al., 2021]



Neural celullar automata for texture synthesis. Really interesting parametrization but generated images are small and lack semantic coherence.

SeamlessGAN





SeamlessGAN sample

SeamlessGAN: Introduction

Training

We adapt an ***adversarial expansion*** *[Zhou et al. 2017]* GAN training scheme.

We modify their framework to make it:

Faster: 5 hours → 40 minutes
Generate ***texture stacks***

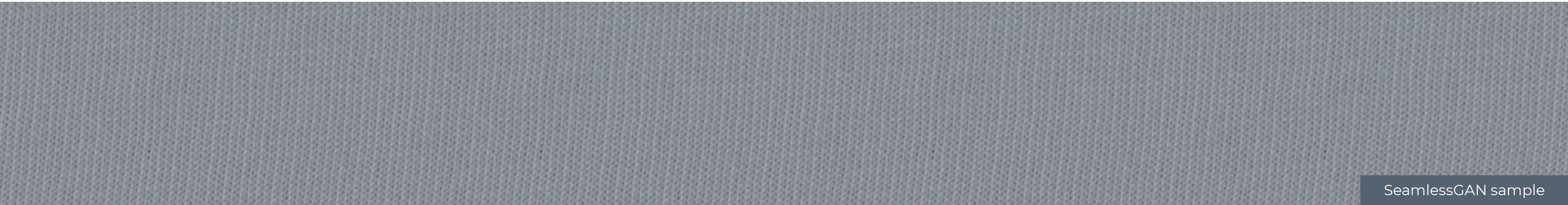
Tiling is not learned during training.

Sampling

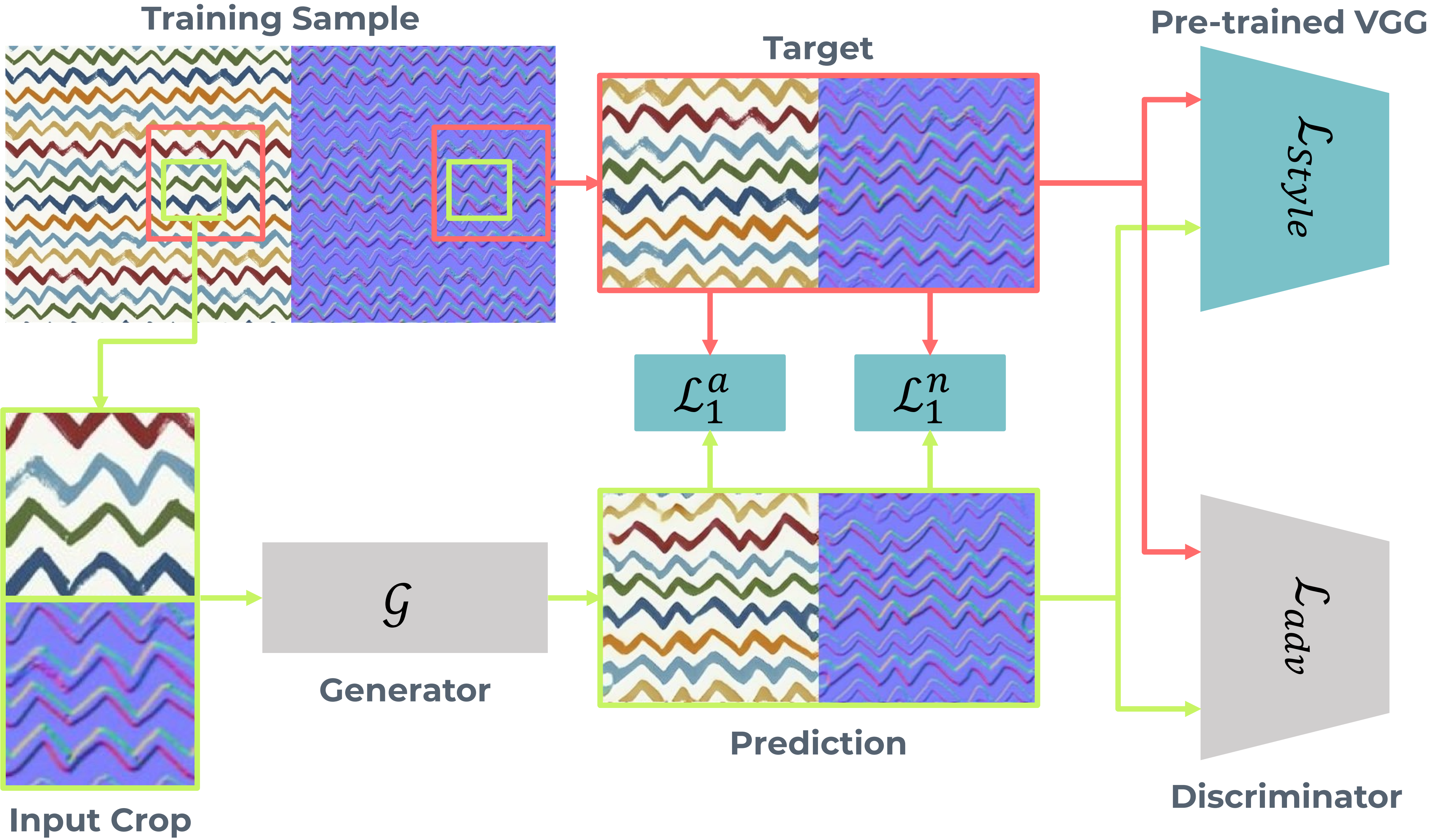
We propose a sampling strategy to generate new **tileable** texture stacks.

- **Latent Space Manipulation** to generate seamlessly tileable textures
- Reuse the **discriminator** as a test-time quality metric

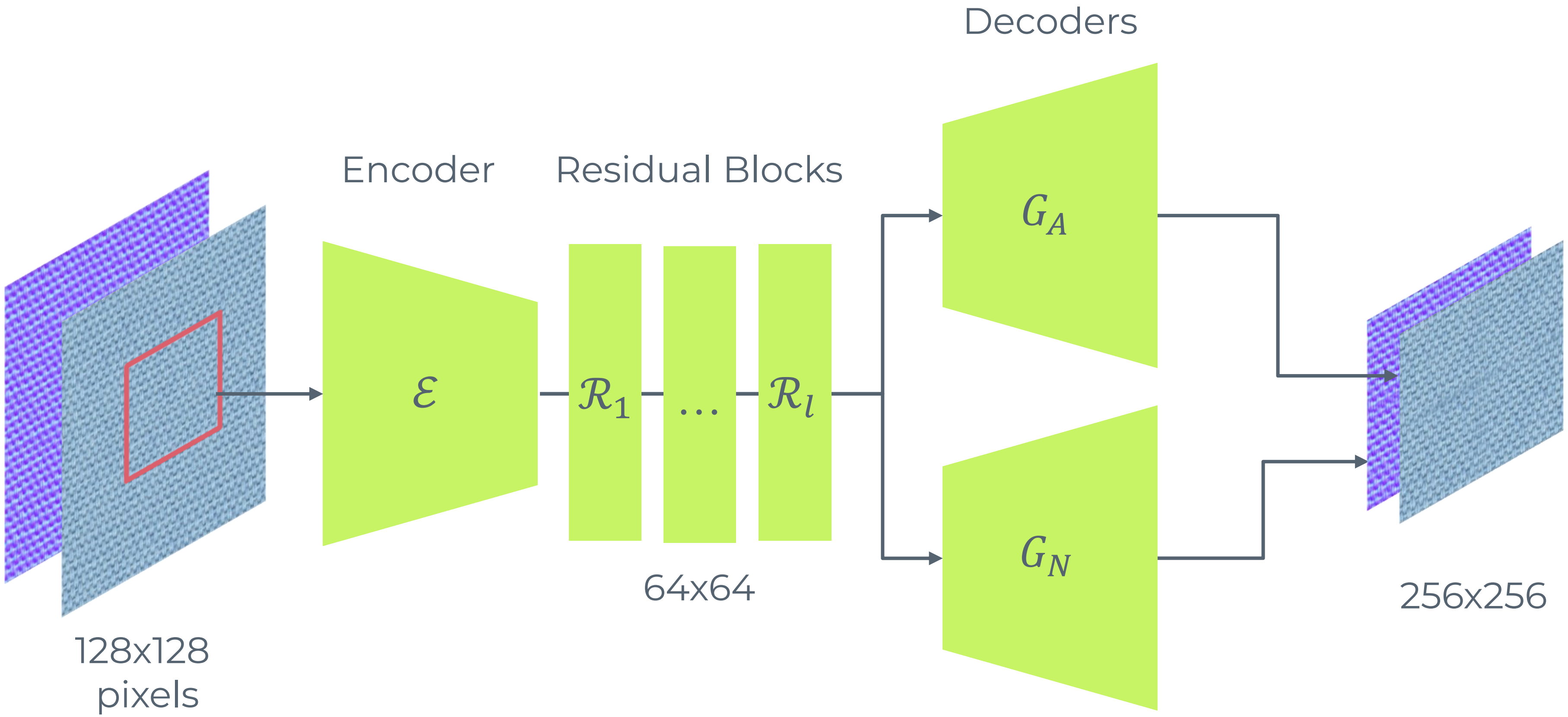
Sampling algorithm to generate multiple tileable textures automatically



SeamlessGAN: Adversarial Expansion Training



SeamlessGAN: Fully-Convolutional Generator



We use the same generator as in [Zhou et al. 2017], with two modifications:

- We use **Instance Normalization** → Better training properties
- We allow for **multiple** texture maps.

SeamlessGAN: PatchGAN Discriminator

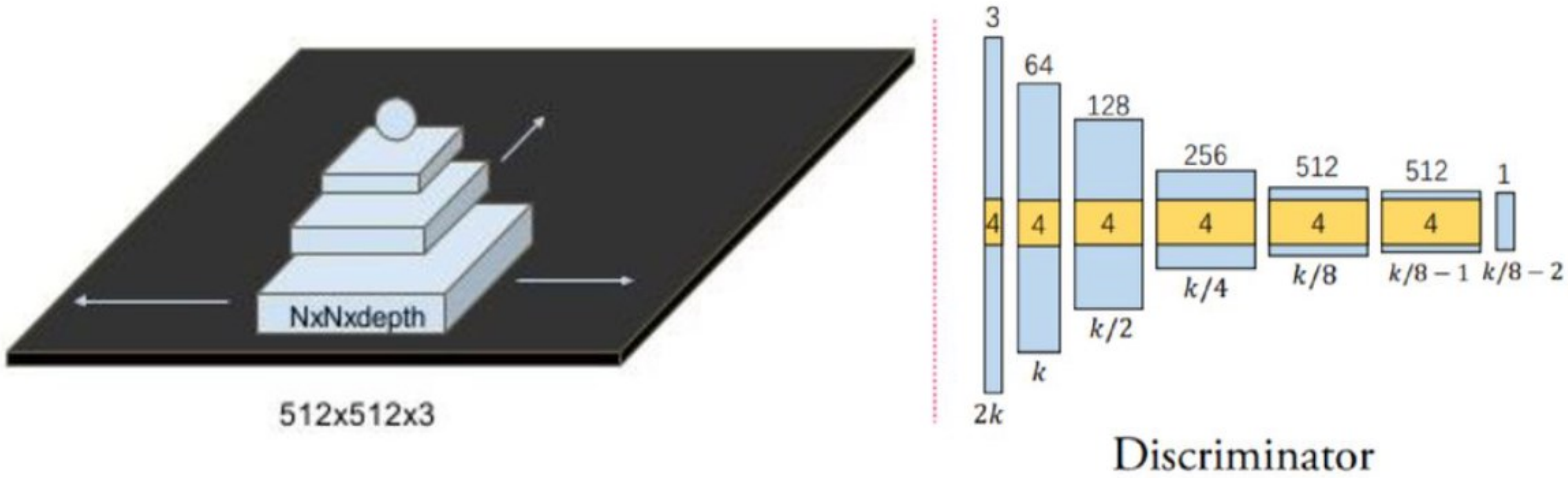


Figure from [Zhou et al. 2017]

PatchGAN discriminators provide real/fake estimations for **patches** of images.
They are used in many image GAN papers (Image-To-Image Translation, VQGAN, etc.).
Great for texture synthesis as they model repetitions naturally.
We can use the discriminator as a test-time quality metric.

SeamlessGAN: Implementation Details

We train one model per texture, for 50000 iterations, with decaying learning rate.

Training takes around **40 minutes** on a 1080Ti GPU

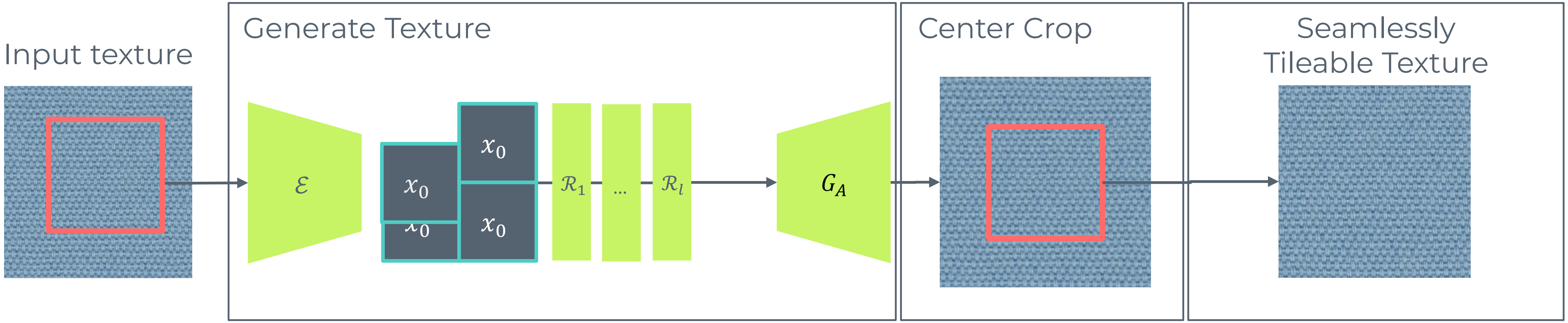
We use **mixed precision training and automatic gradient scaling** → Faster training and less memory consumption.

Implementation done fully in **Pytorch**.

We train and test on images of >400 pixels.

SeamlessGAN sample

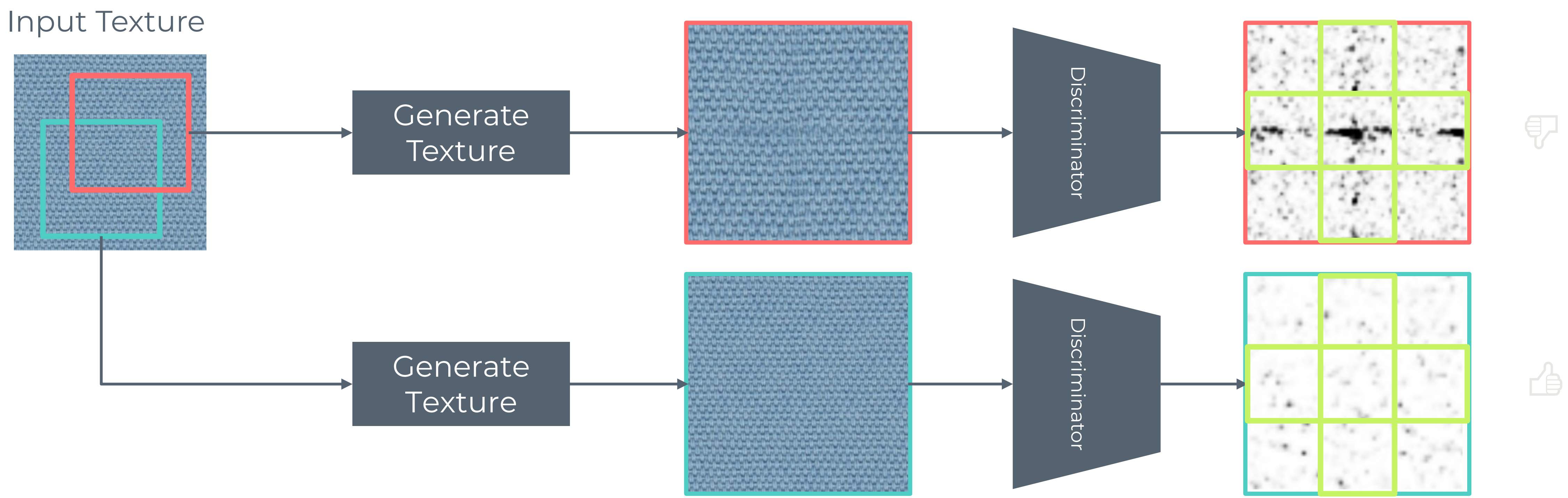
SeamlessGAN: Sampling 1 Tileable Texture



To generate a tileable texture from a **pre-trained** generator, we:

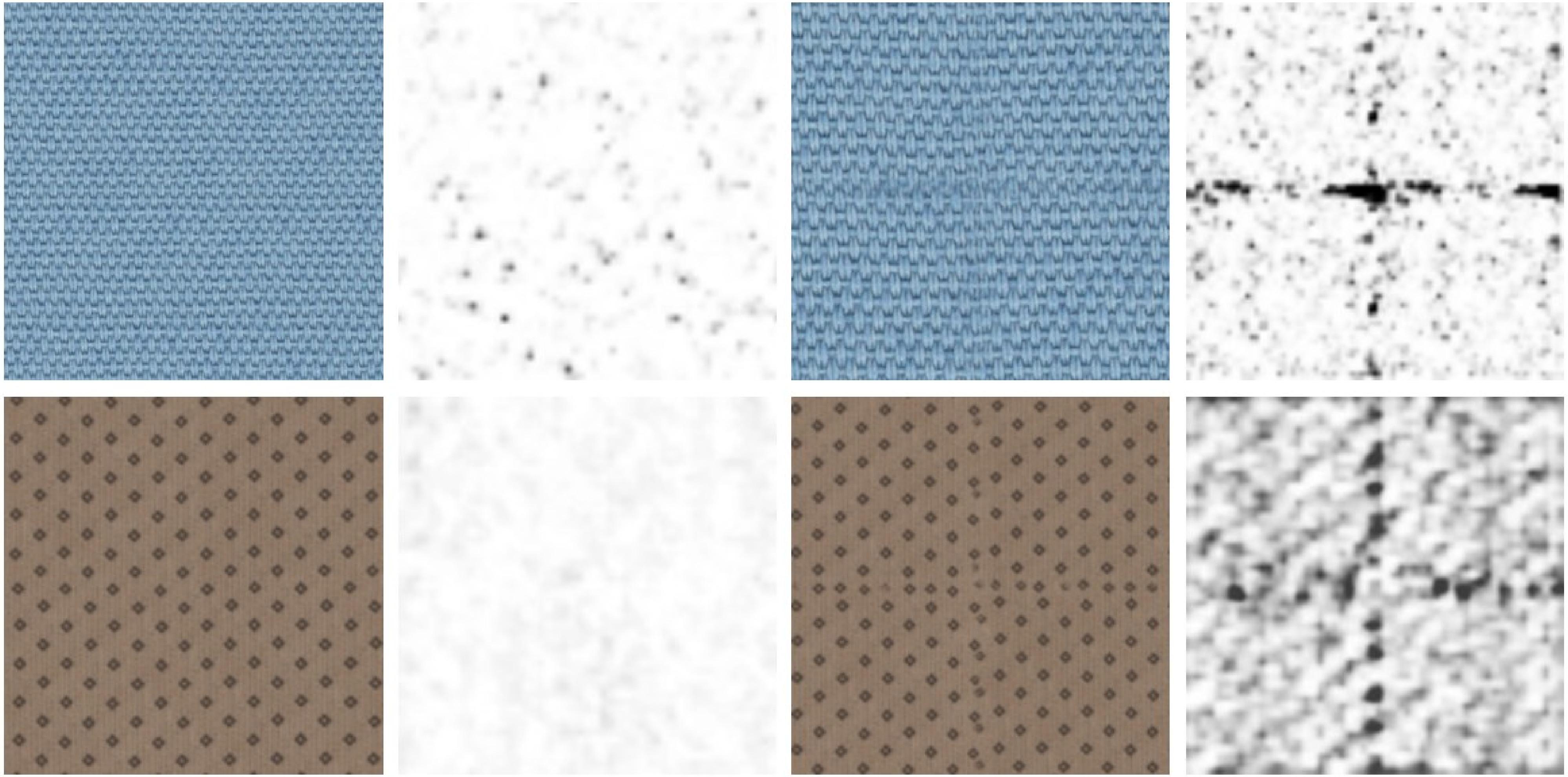
- Randomly select a crop of the texture
- Get its latent representation
- **Vertically and horizontally concatenate this representation**
- Pass it through the rest of the generator
- Crop the center of the output

SeamlessGAN: Discriminator as Quality Function



Not all textures are created equally. Depending on the input crop selection, the latent space manipulation may work or not.
We can use the **discriminator** to detect artifacts on the generated textures.
The discriminator provides learned local estimations of the quality of the textures.
Tiling artifacts appear on the central areas, where the latent space was concatenated with itself.
We can use those areas to detect tiling artifacts.

SeamlessGAN: Sampling – Discriminator as Quality Metric



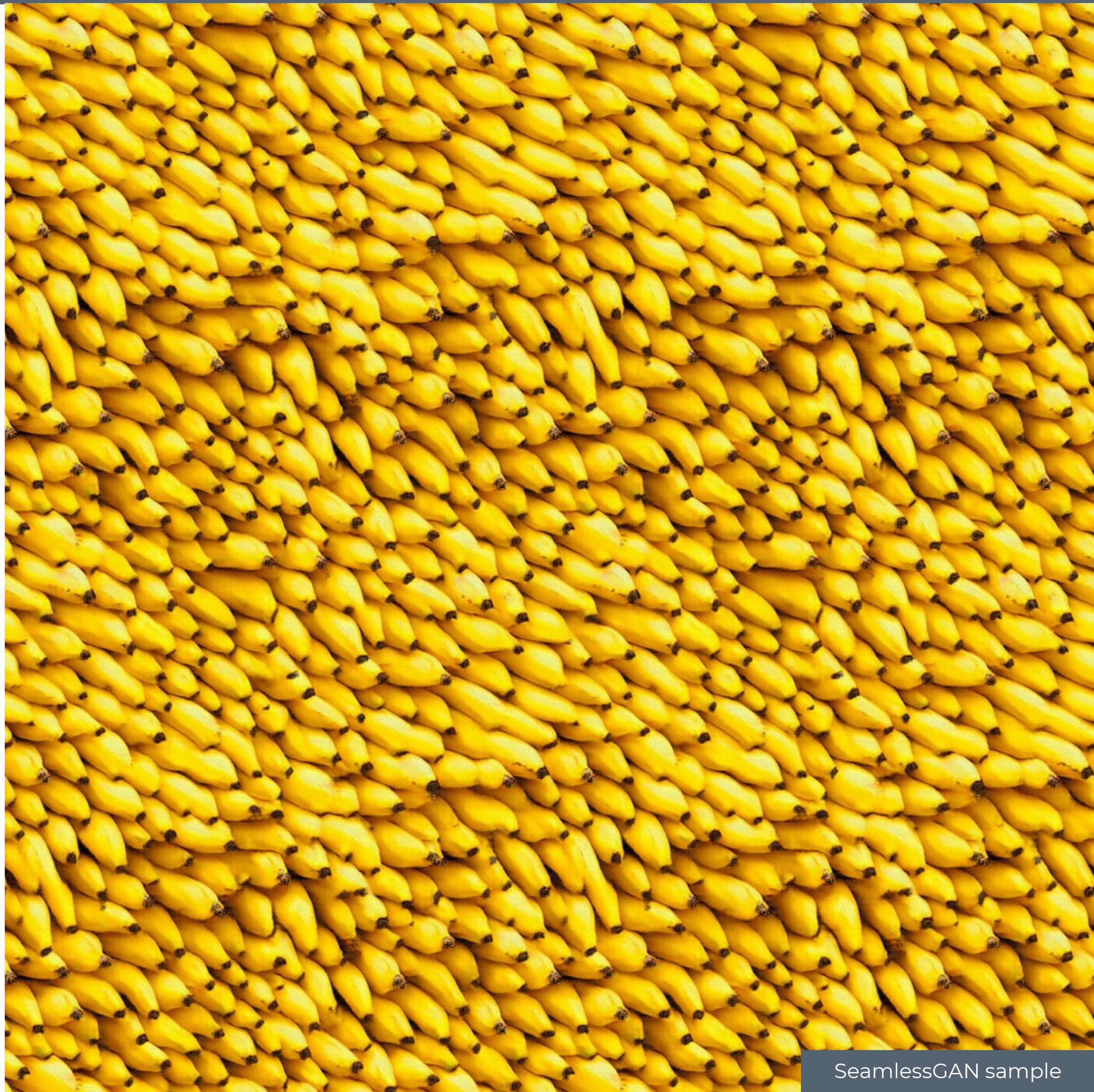
Evaluation

Impact of loss function

Impact of network architecture

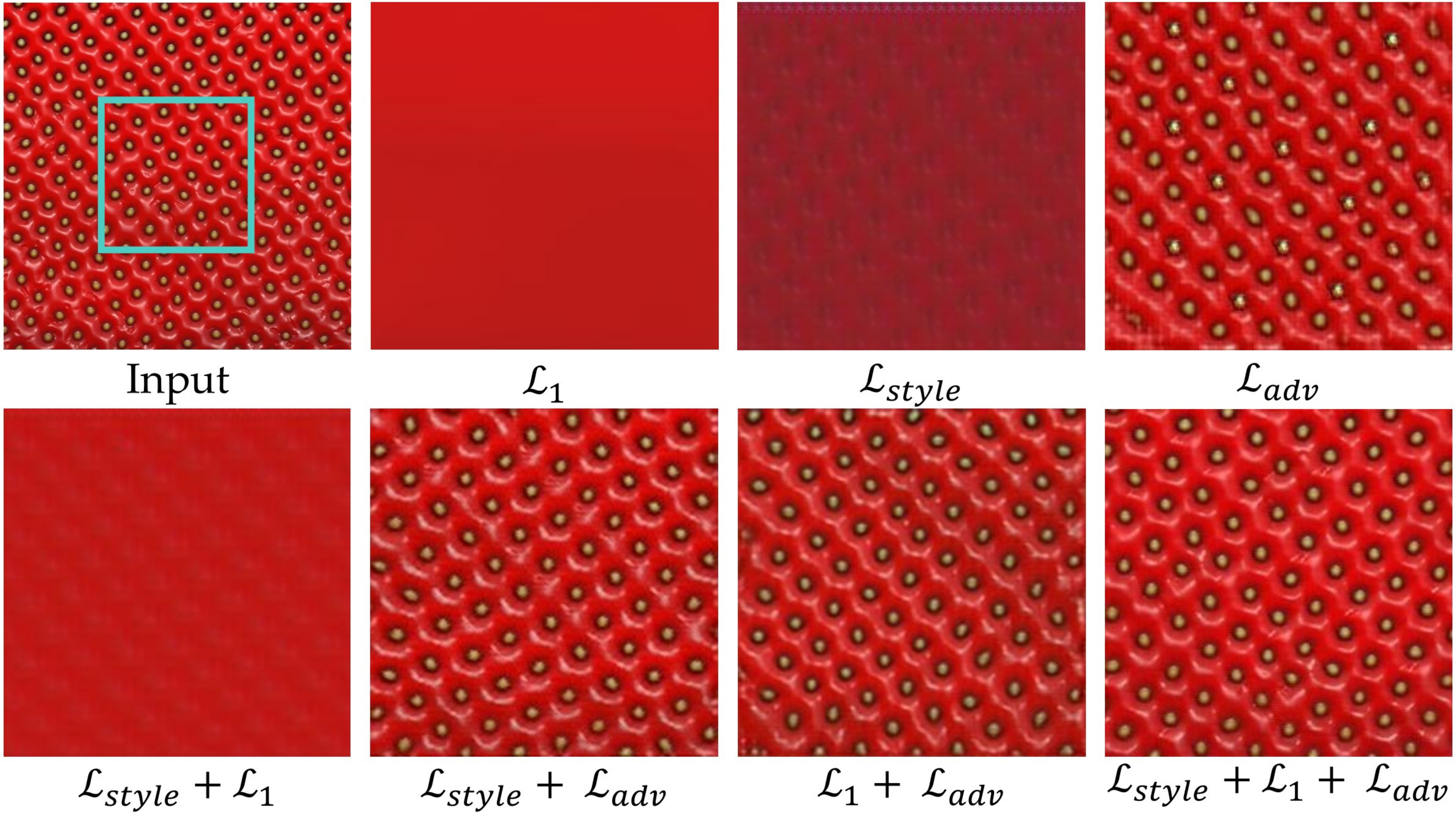
- Network size
- Network design

Where to tile the latent space?

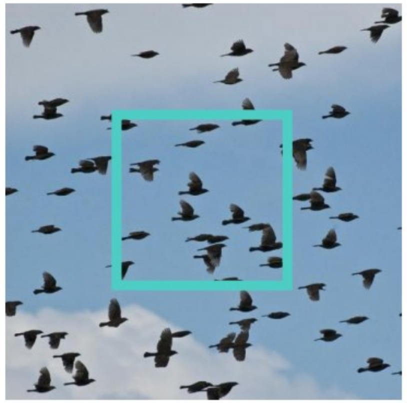


SeamlessGAN sample

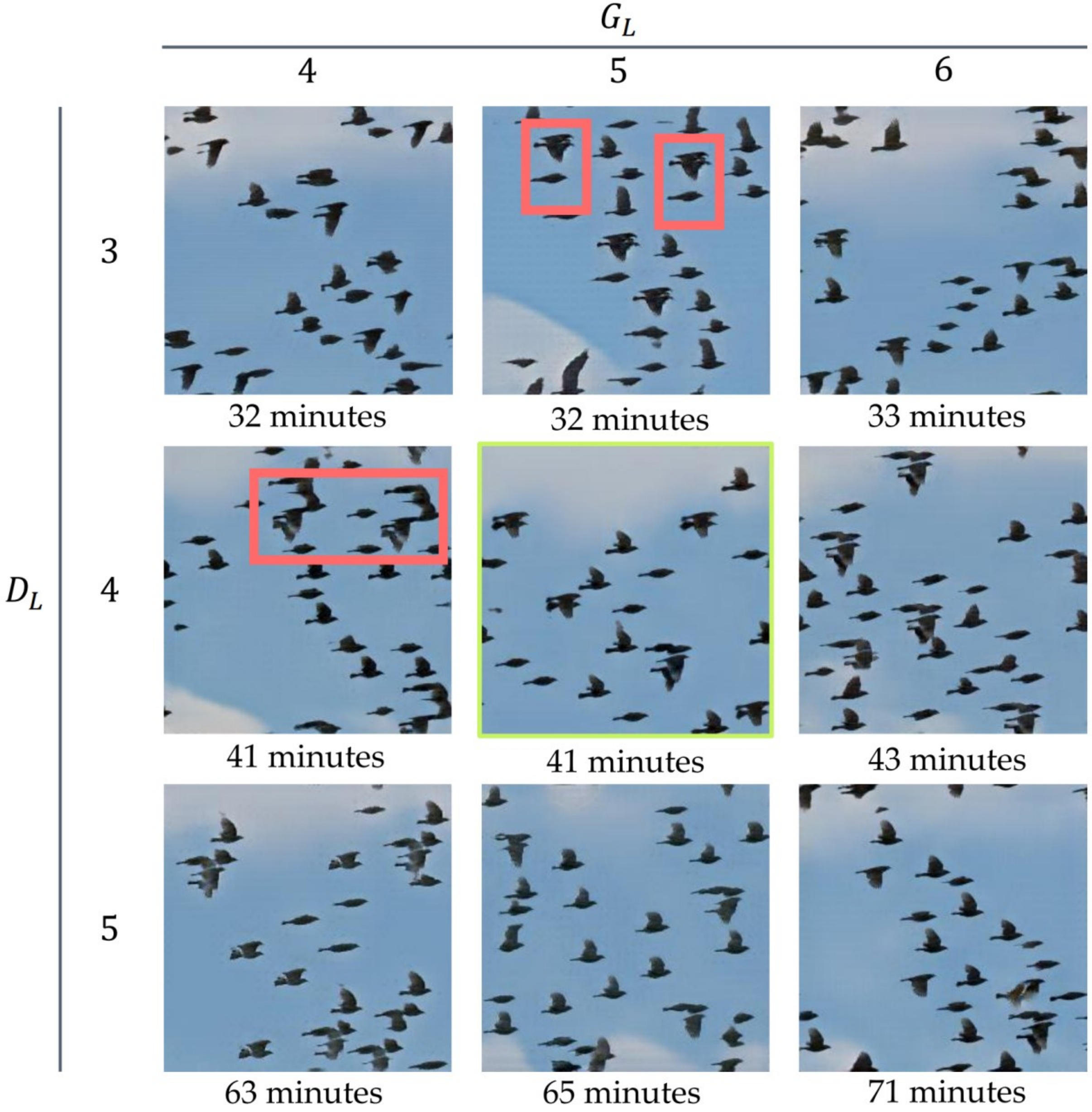
Evaluation: Loss Function



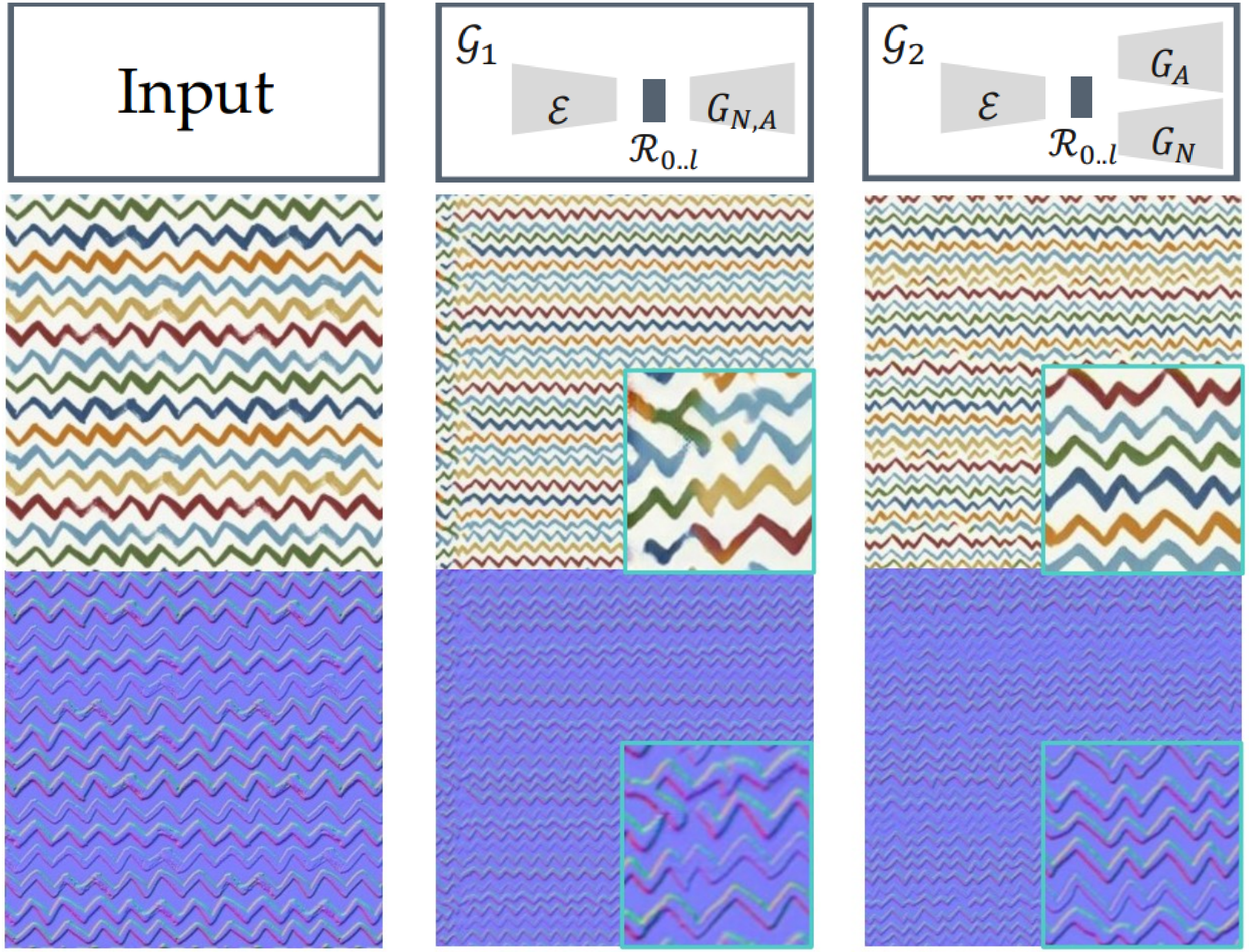
Evaluation: Network Size



Input

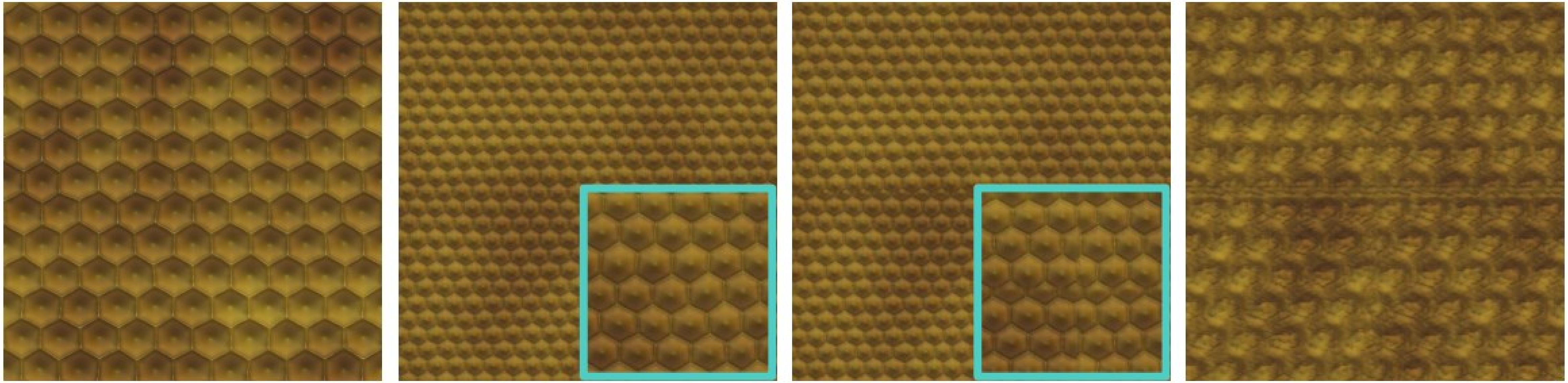


Evaluation: Network Architecture Design



Separate decoders on the generator provide better texture maps.

Evaluation: Which latent space to tile?

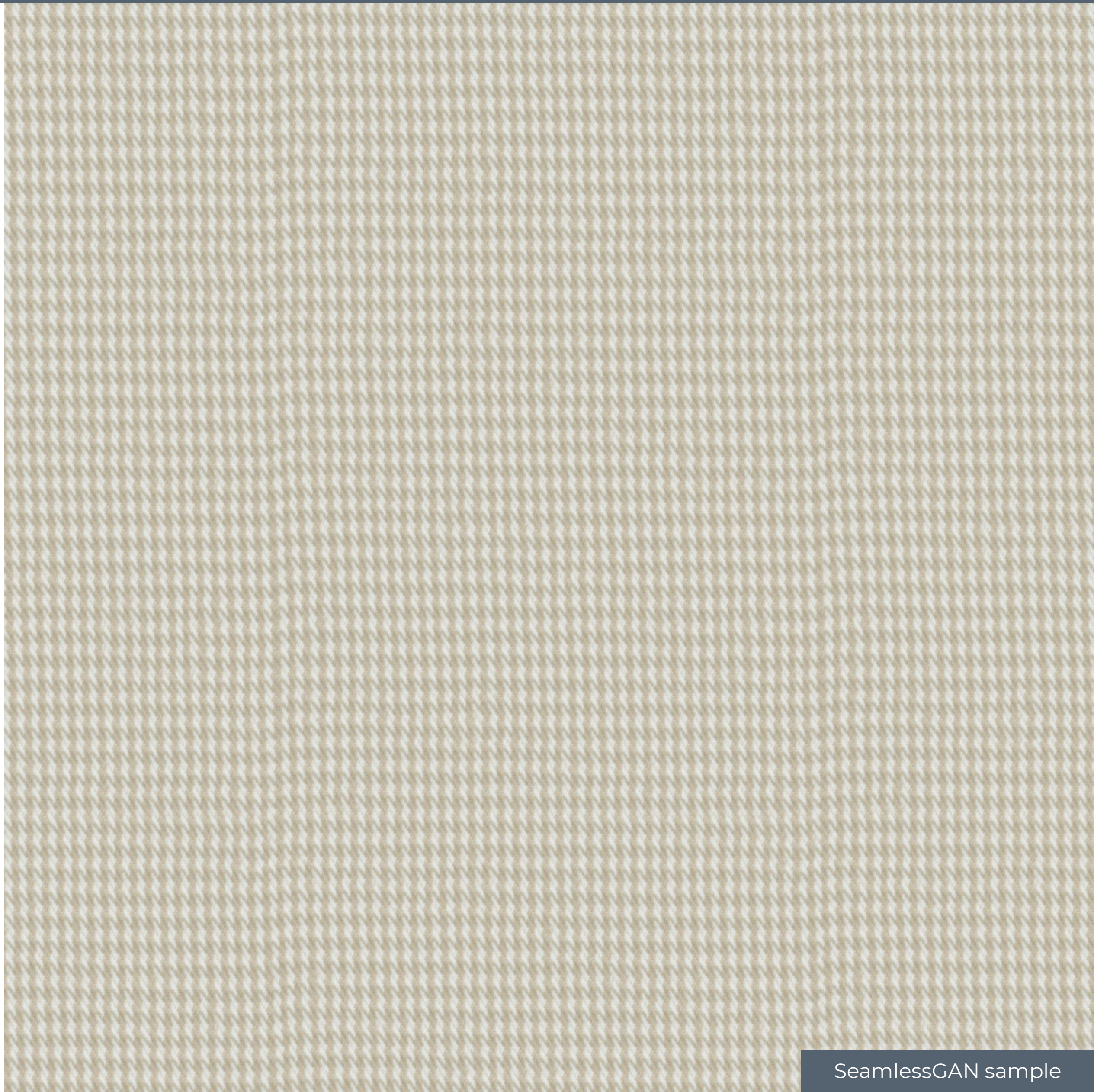


(a) t (b) $\hat{\mathcal{T}}, l = 0$ (c) $\hat{\mathcal{T}}, l = 3$ (d) $\hat{\mathcal{T}}, l = 5$

Tiling the first residual layer provides the best results overall.

Results

- Qualitative analysis
- Multiple outputs from a single sample
- Comparisons with previous work
- Failure cases
- Limitations

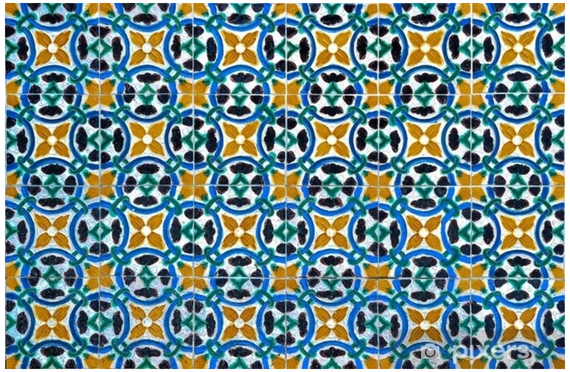


SeamlessGAN sample

Results: Qualitative Analysis - Videos

Input Texture

Input Texture



Input Texture

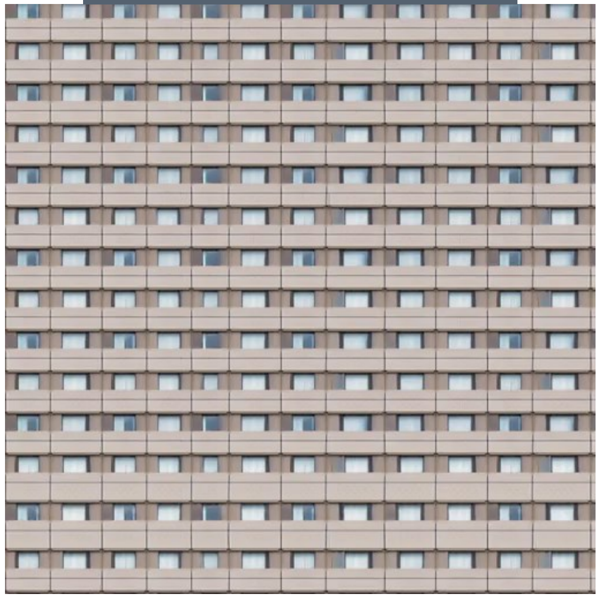
Input Texture

Tiled Output

Tiled Output

Tiled Output

Tiled Output

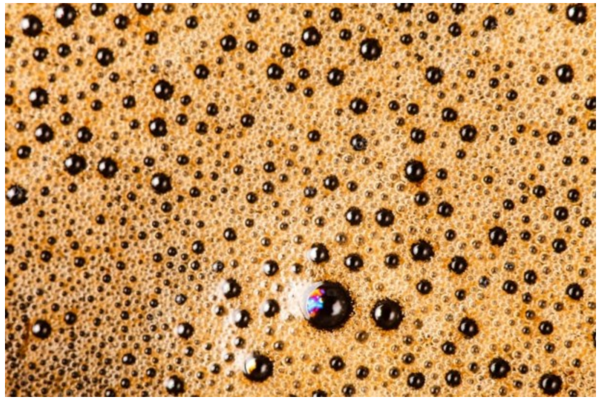


Results: Qualitative Analysis - Videos

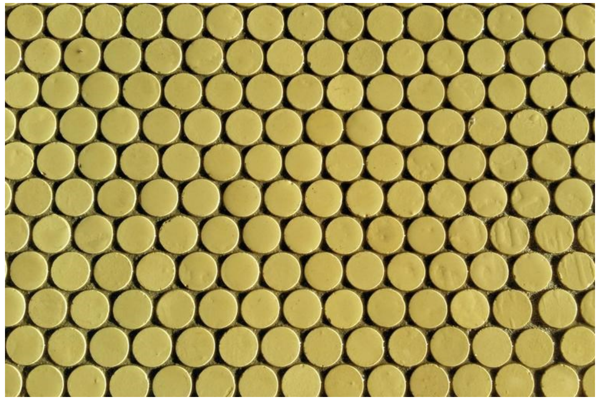
Input Texture

Input Texture

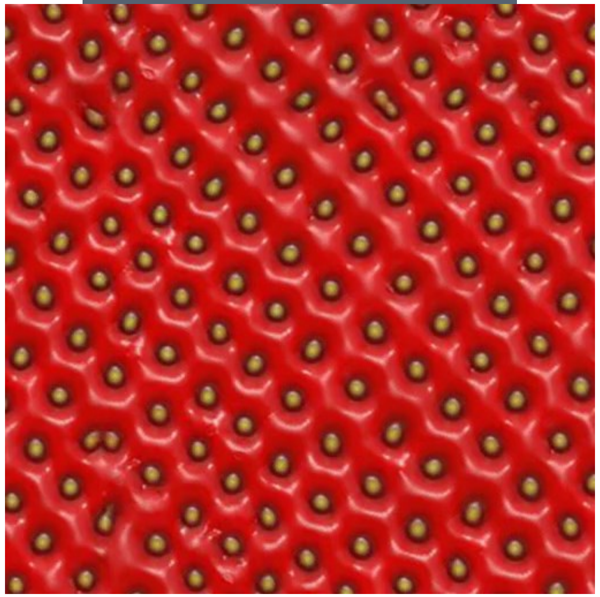
Input Texture



Input Texture

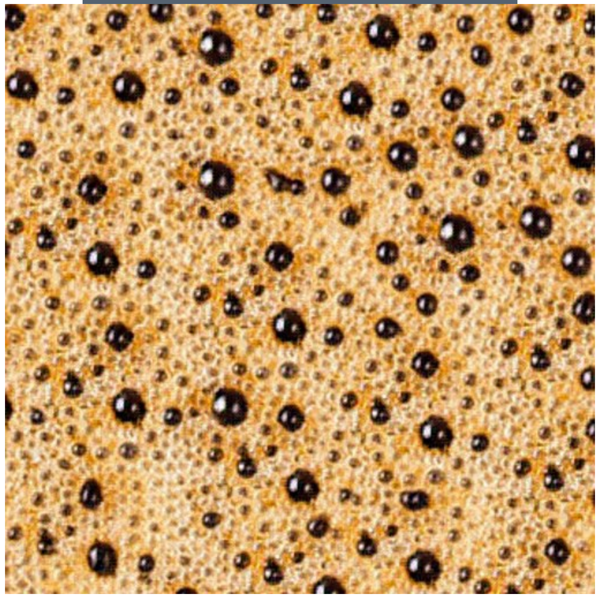


Tiled Output

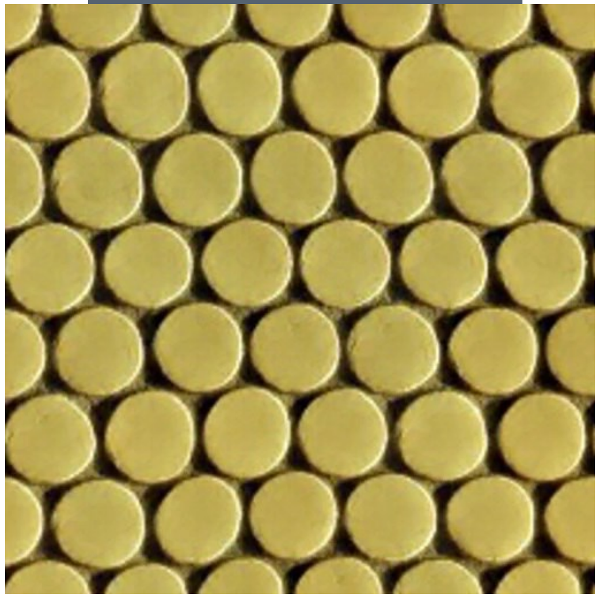


Tiled Output

Tiled Output

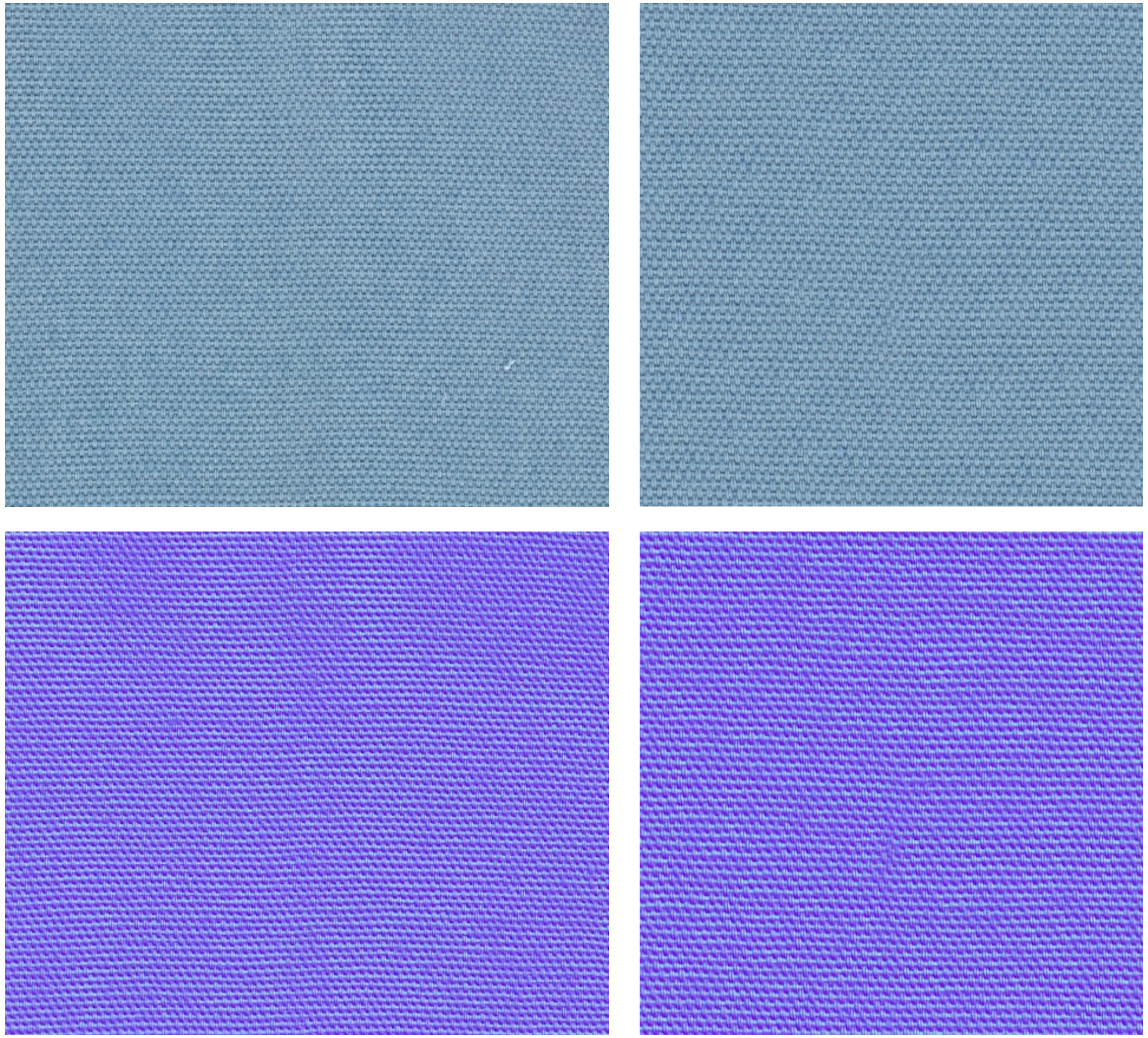


Tiled Output

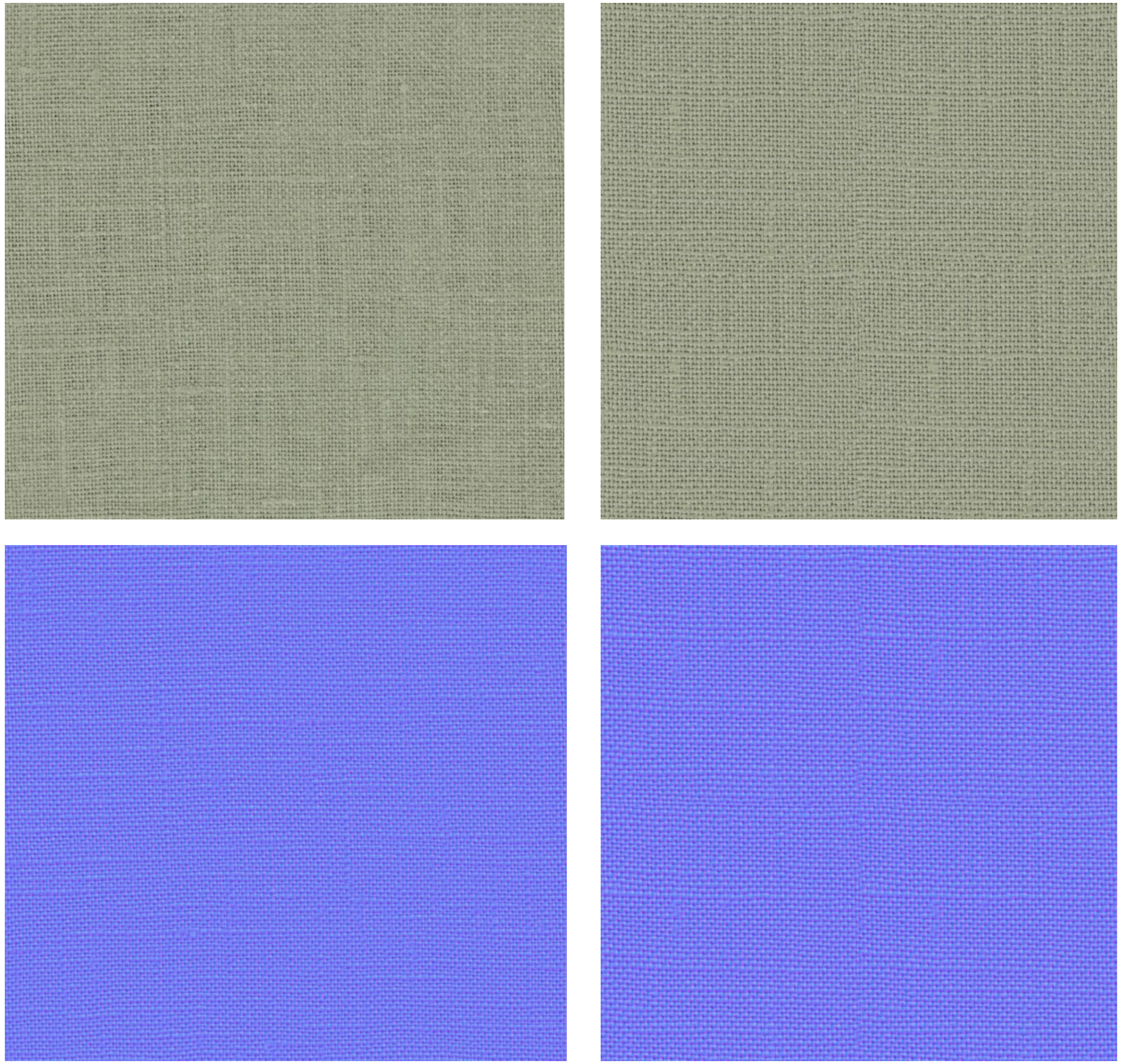


Results: Qualitative Analysis – Multi Layer

Input Textures | **Output**

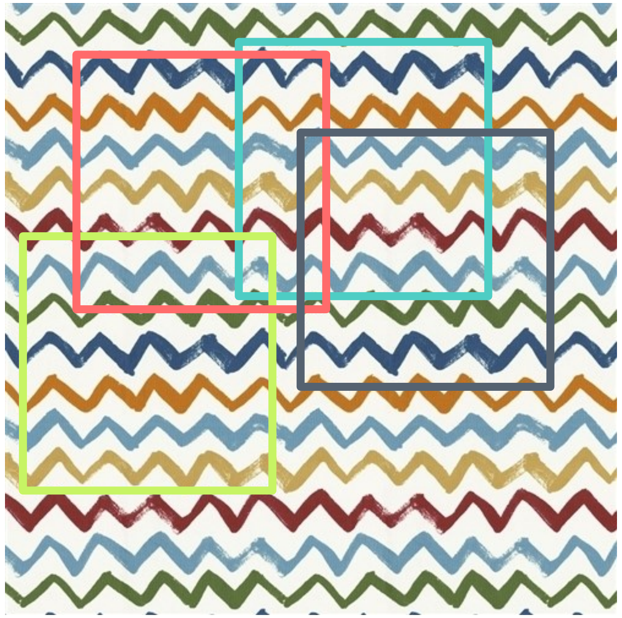


Input Textures | **Output**

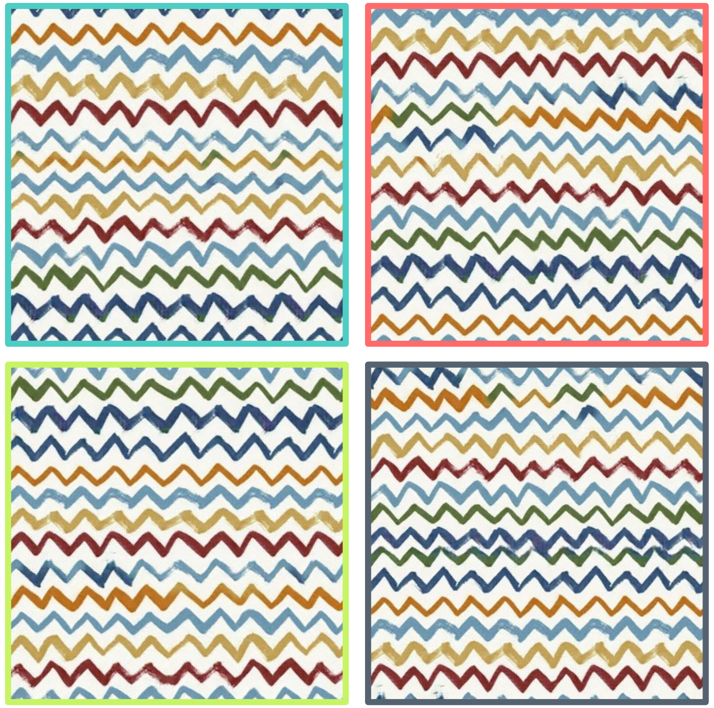


Results: Multiple Outputs

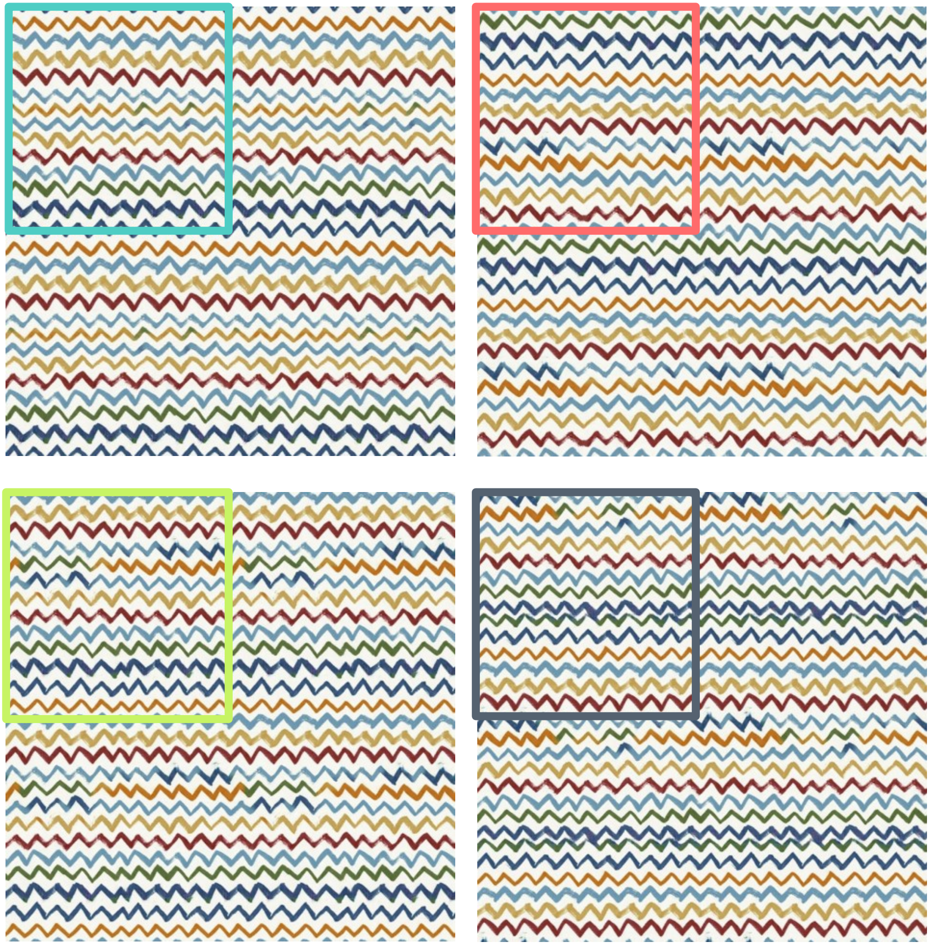
Input crops



Outputs

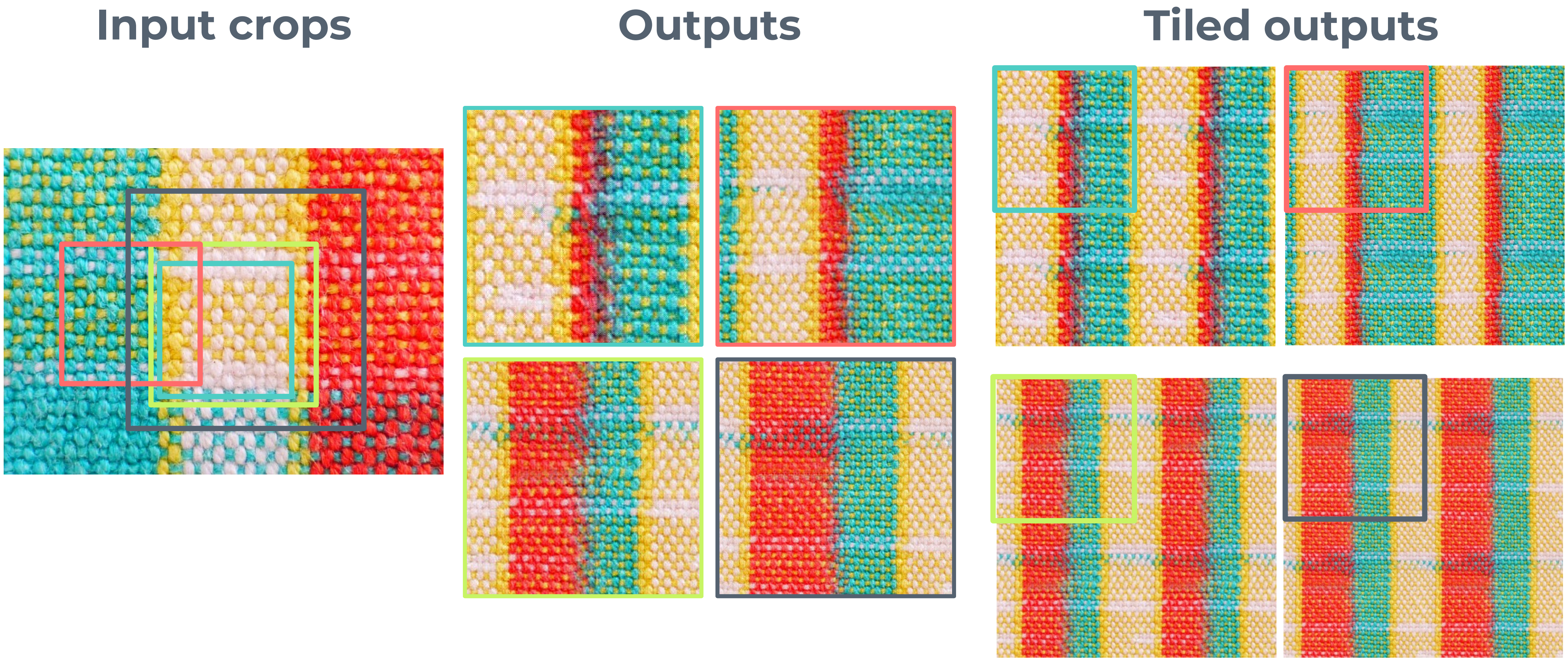


Tiled outputs



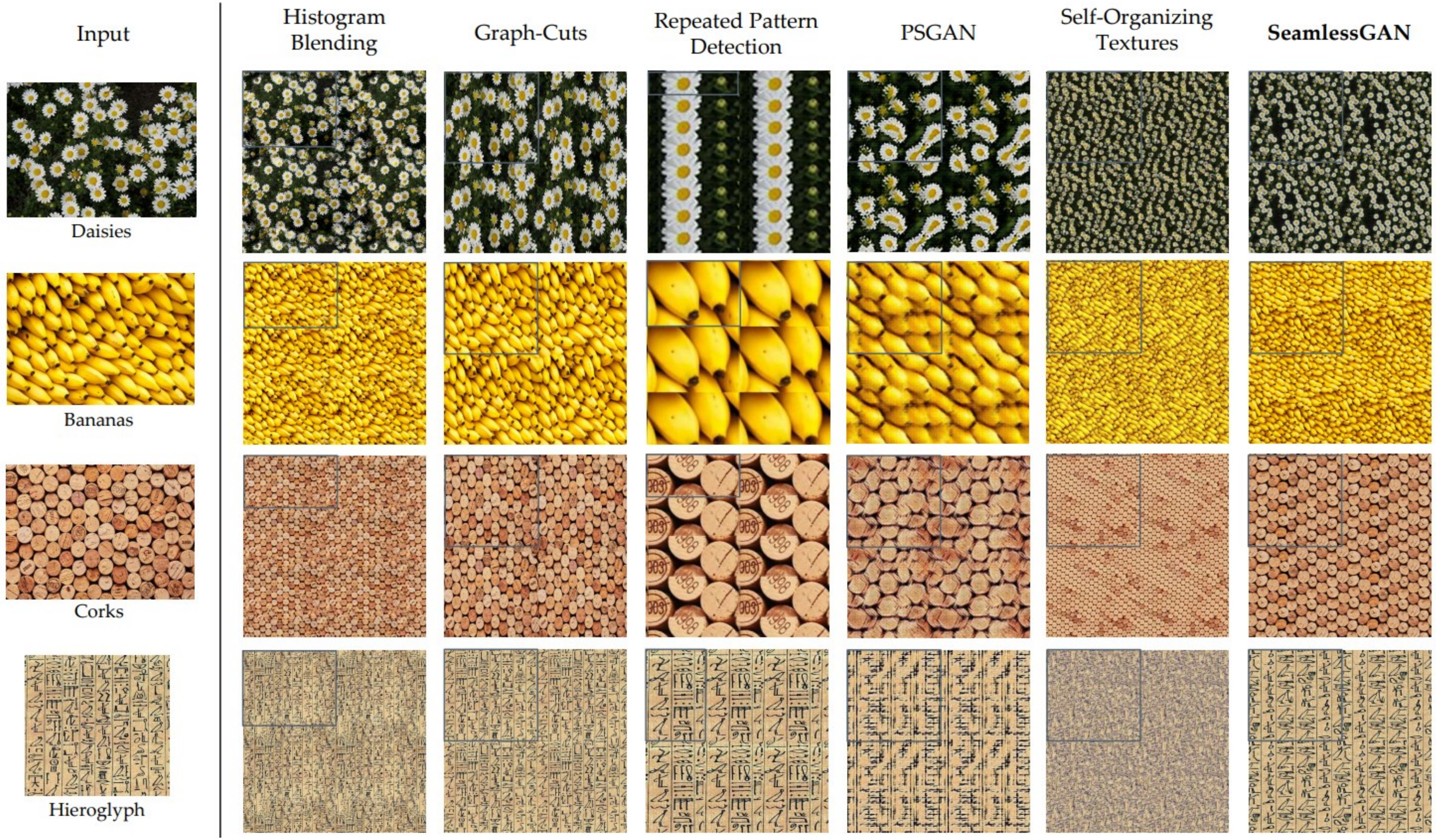
SeamlessGAN allows for generating a variety of outputs from a single input.

Results: Multiple Outputs

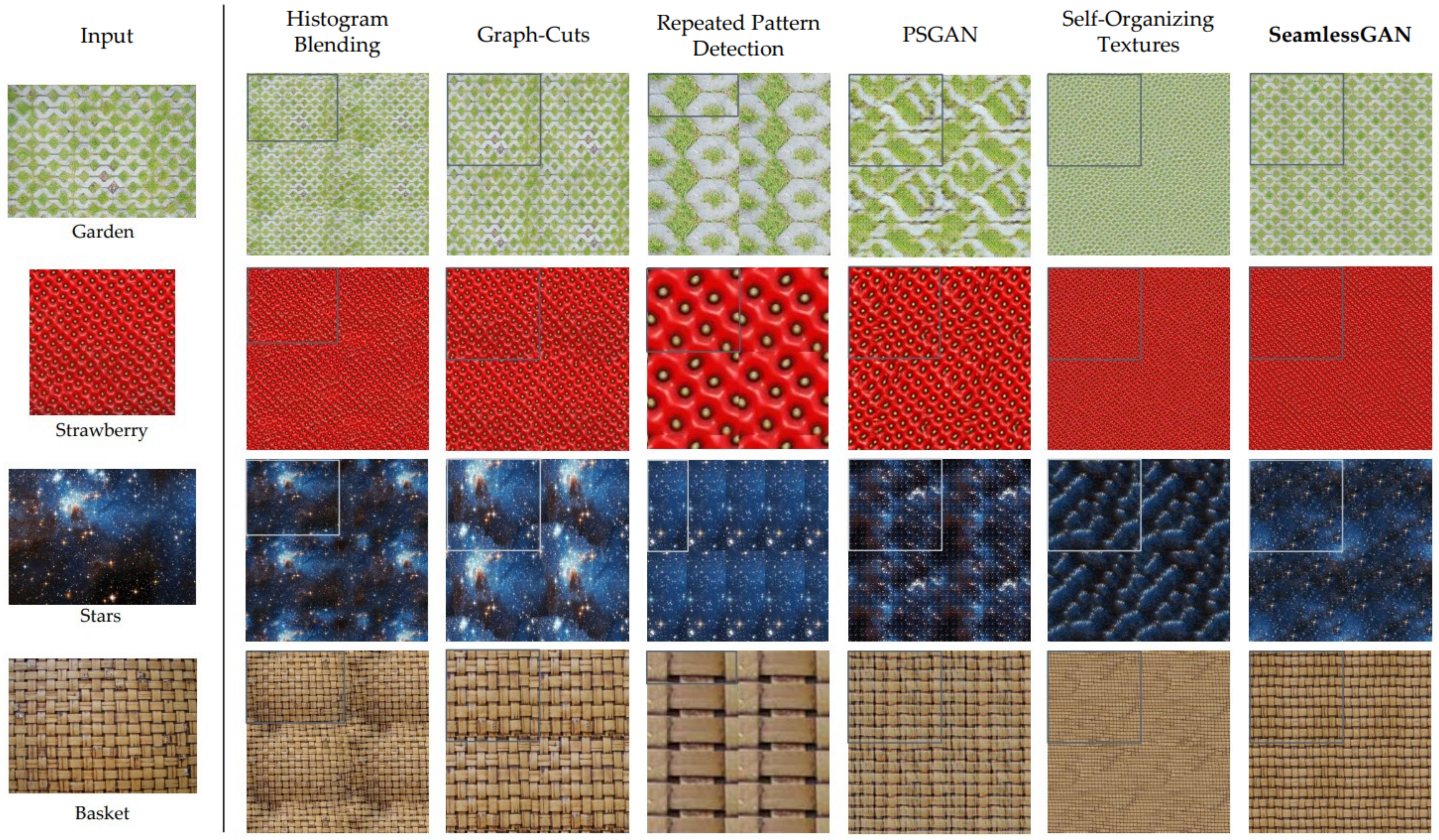


The different outputs can be of different sizes.

Results: Comparisons with Previous Work

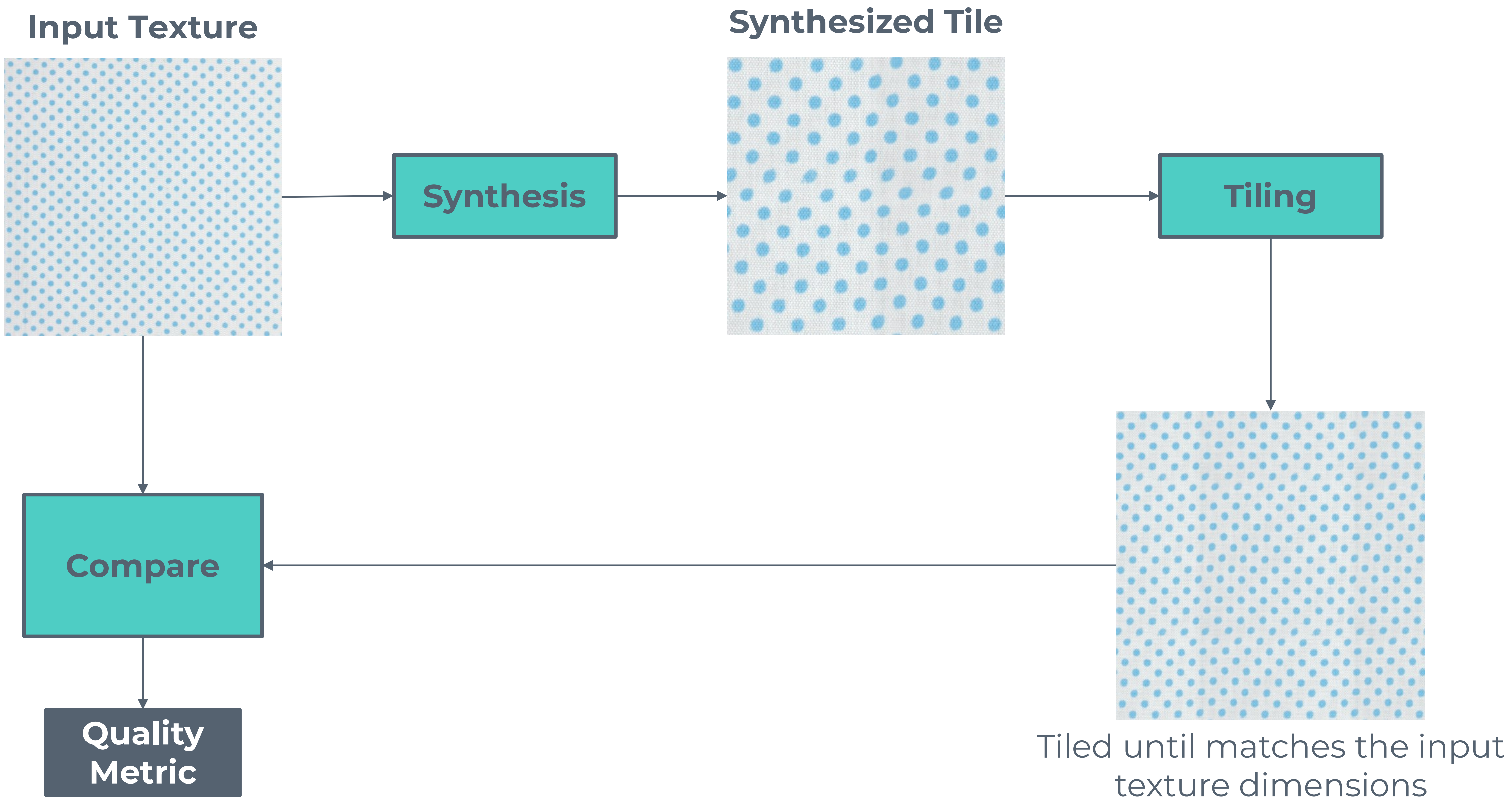


Results: Comparisons with Previous Work



*More results on the supplementary material

Results: Quantitative Comparison



Results: Quantitative Comparison

Structural Similarity Index Measure:
Measures structural similarity between two images

Single-Image Fréchet Inception Distance:
Measures quality of GAN-generated images

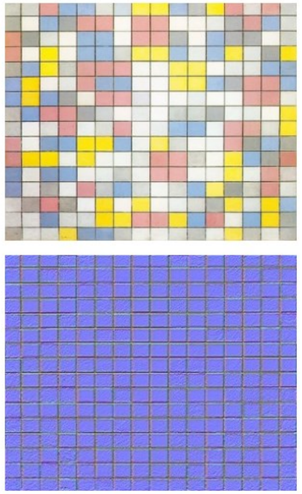
Learned Perceptual Image Patch Similarity:
Uses CNNs to measure distance between two images

	SSIM ↑	Si-FID ↓	LPIPS ↓
Deloit *et al.* [20]	0.1424	1.3471	0.6207
Li *et al.* [22]	0.2086	0.9529	0.5818
Moritz *et al.* [13]	0.1968	0.7620	0.5171
Rodriguez *et al.* [21]	0.2144	1.2958	0.5137
Bergmann *et al.* [16]	0.1723	1.4355	0.5624
Niklasson *et al.* [23]	0.1753	1.3171	0.5328
SeamlessGAN	**0.2341**	**0.6311**	**0.4792**

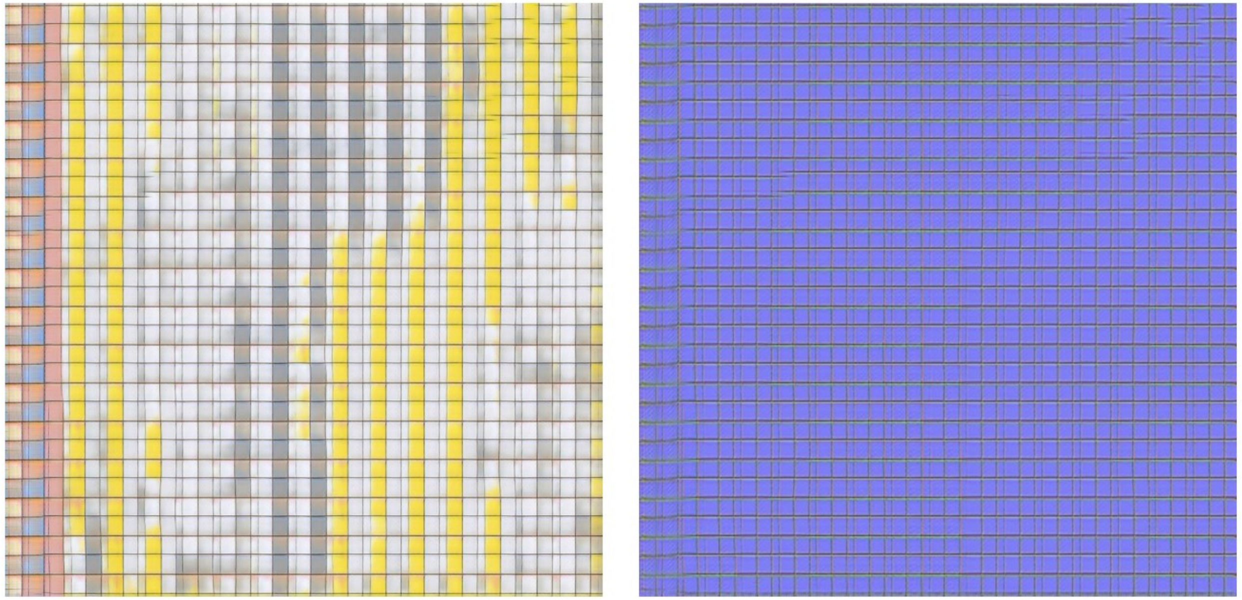
SeamlessGAN outperforms non-parametric and parametric models

Results: Failure Cases

Input data



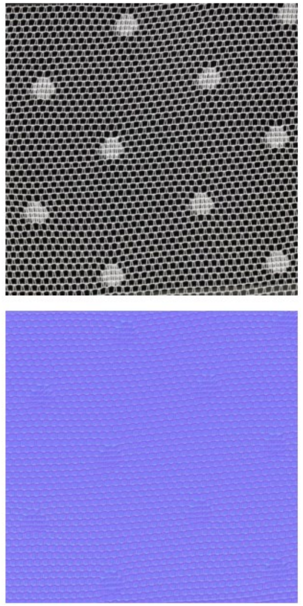
Outputs



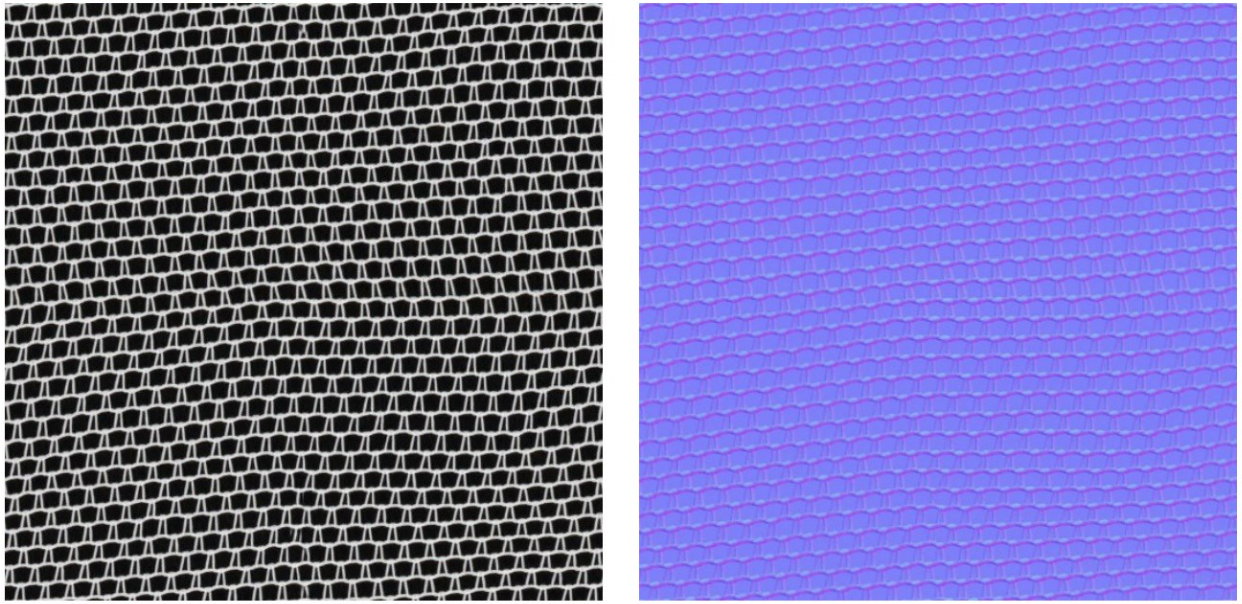
SeamlessGAN fails when there is no obvious way of synthesizing the texture

Results: Failure Cases

Input data



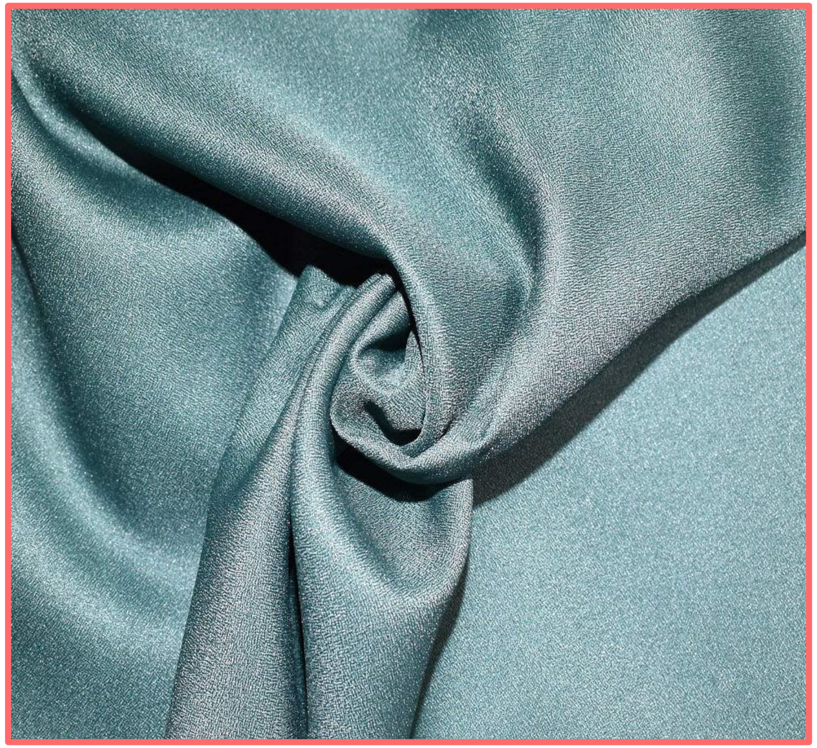
Outputs



SeamlessGAN fails to preserve important but unfrequent details

Results: Limitations

Seamless borders may still produce visible repetitions



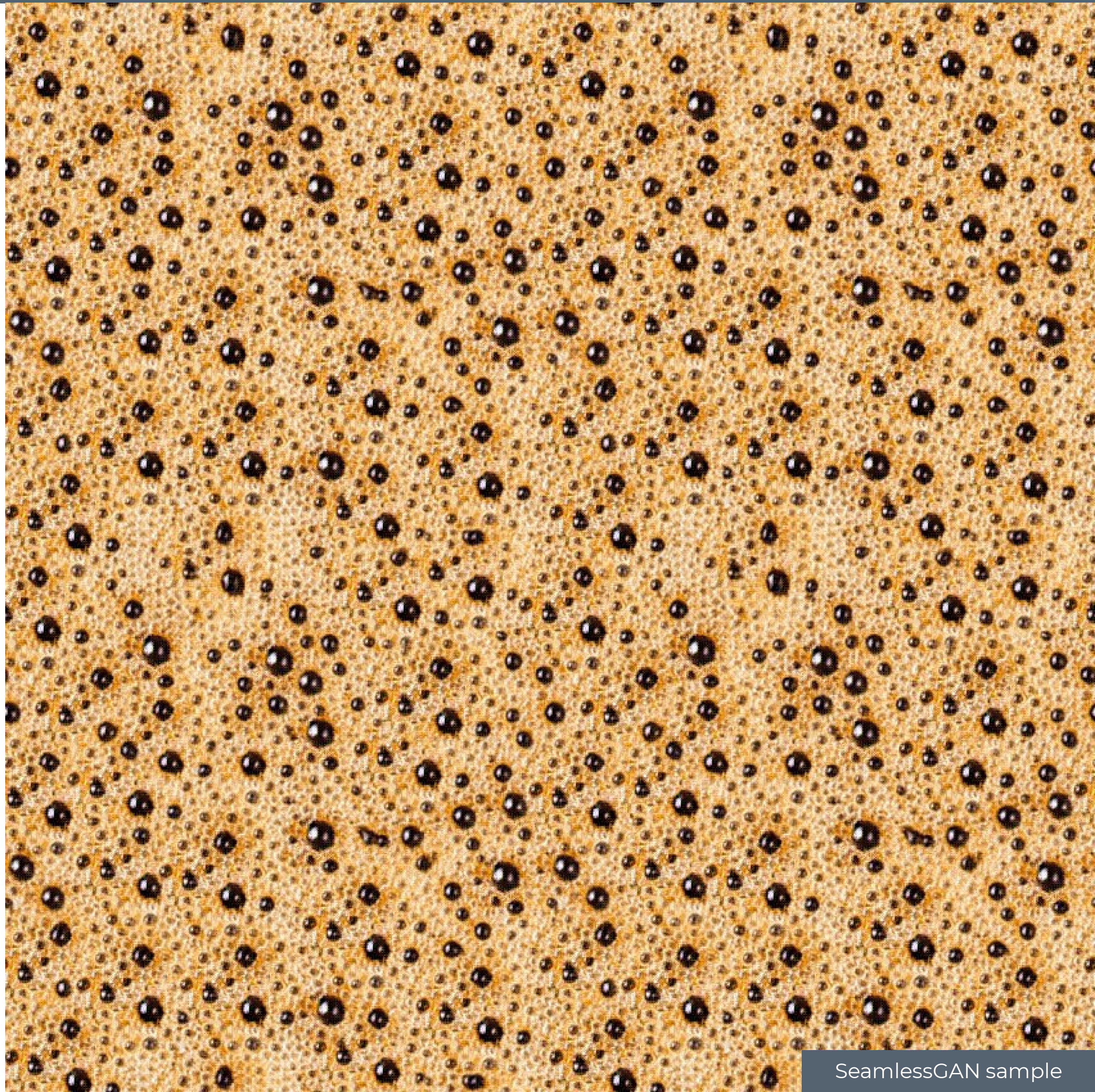
SeamlessGAN needs *frontoplanar* inputs

SeamlessGAN needs inputs with a single texture

Summary

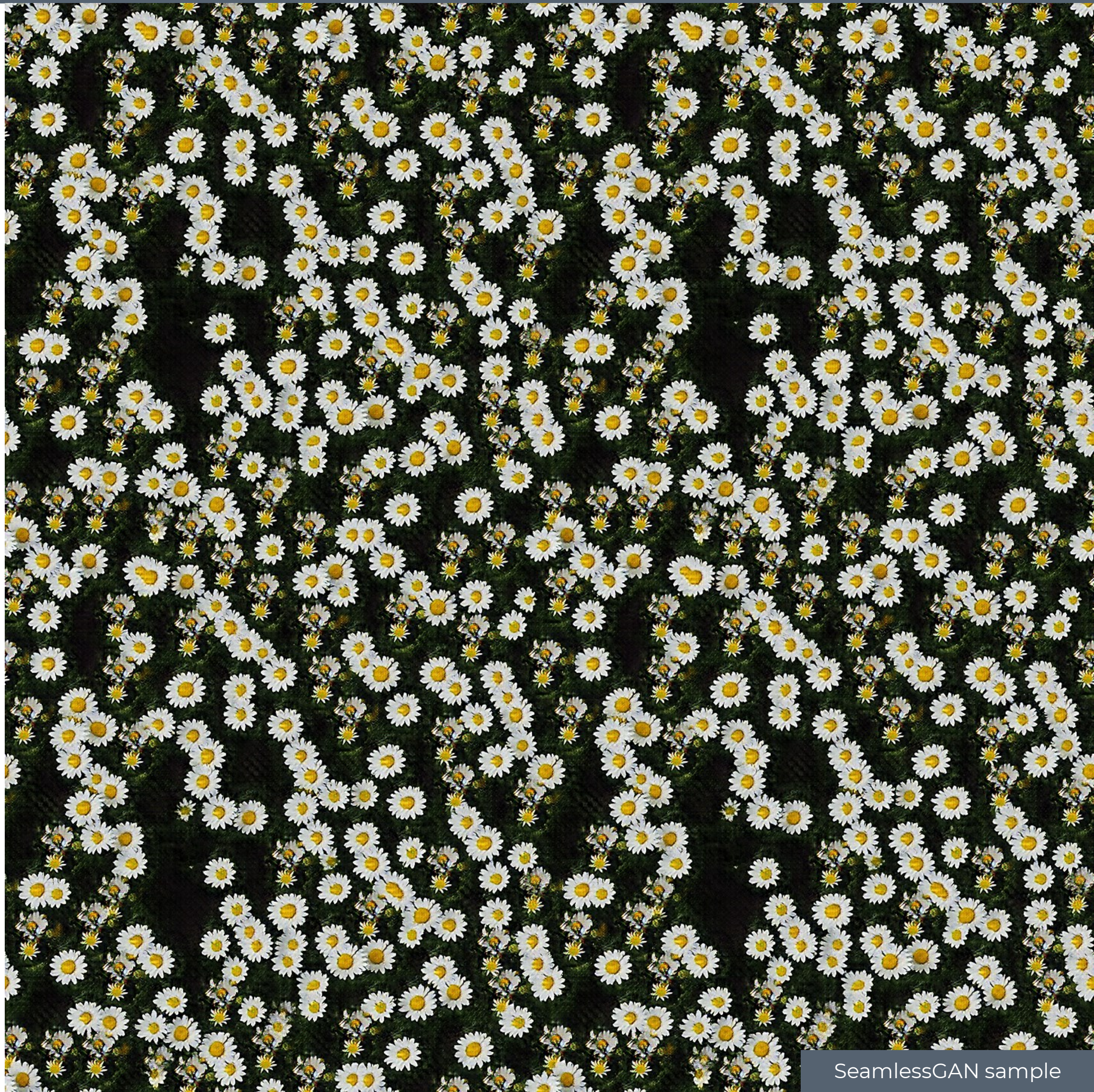
We presented **SeamlessGAN,** a single-image generative model capable of generating tileable texture maps.

- Provides better results than previous work
- Is more efficient than previous work
- Can handle multiple maps at once
- Works with a large variety of images



Future Work

- **Image pre-processing**
- **Architectural improvements:**
  - Attention-based mechanisms
  - U-Net shaped discriminators may provide more accuate
- **Loss function improvements:**
  - Fourier-space losses
  - Improved perceptual loss
- **Learn to generate tileable textures from scratch**
- **Move beyond single-image GANs**
  - Pre-trained GANs
  - Diffusion models



SeamlessGAN sample

Further Information

Published in Transactions on Visualization and Computer Graphics (TVCG).

- 📄Paper, supplementary material and more results in: https://carlosrodriguezpardo.es/projects/SeamlessGAN/
- 🤖 Reddit post: https://www.reddit.com/r/MachineLearning/comments/t78zoq/r_seamlessgan_selfsupervised_synthesis_of/
- 🐦Twitter thread: https://twitter.com/Crp94/status/1481916191459446784
- **To be presented at AI For Content Creation Workshop** @ CVPR2022 https://ai4cc.net/

SeamlessGAN: Self-Supervised Synthesis of Tileable Texture Maps

IEEE Transactions on Visualization and Computer Graphics (TVCG), 10.1109/TVCG.2022.3143615

Carlos Rodríguez-Pardo[1,2], Elena Garcés[1,3]

SEDDI [1]

uc3m | Universidad Carlos III de Madrid [2]

Universidad Rey Juan Carlos [3]