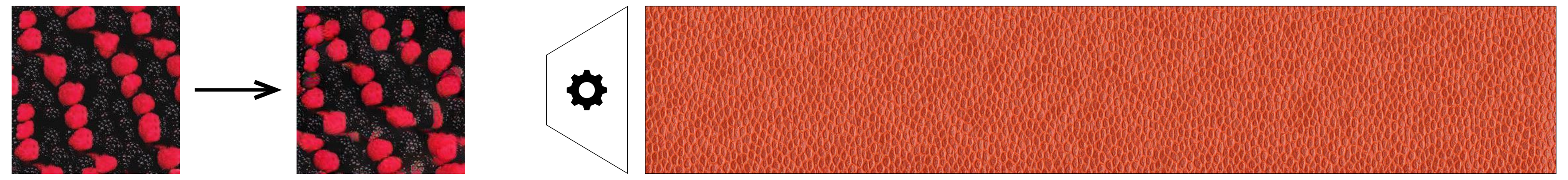
A Sliced Wasserstein Loss for Neural Texture Synthesis

Eric Heitz, Kenneth Vanhoey, Thomas Chambon and Laurent Belcour

Context: Neural Texture Synthesis and Style Transfer

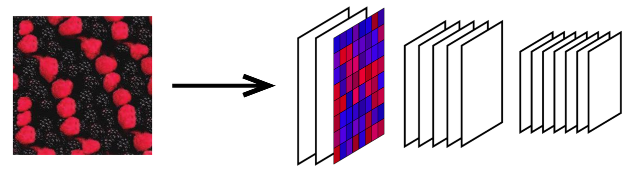
Neural Texture Synthesis (NTS) has two direct applications:



(1) Optimizing a new texture *(2) Training a texture generator*

NTS relies on a loss function to capture distance in texture space.

The rich space of deep feature activations is typically used to compute distances in.

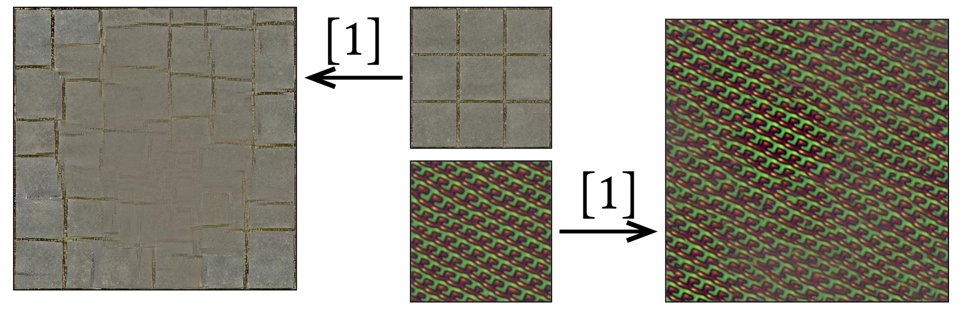


A good loss should capture *complete* stationary statistics of deep feature activations, be *simple* to implement and *tractable* to run.

Existing Loss Functions for Neural Texture Synthesis

1. **Second-order statistics**

Gram-Matrix loss $\mathcal{L}_{\mathrm{Gram}}$ [1] captures correlation between feature channels.

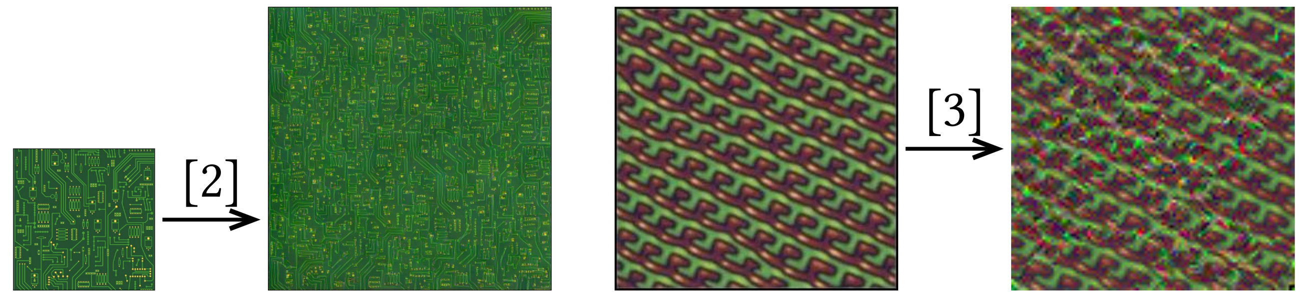


Incomplete

Insufficiently captures texture (e.g., contrast).

2. **Full histogram (i.e., any-order statistics)**

Optimal transport losses capture the full histogram of feature distribution.



Bad complexity

Requires approximations.

Fails on NTS.

3. **Mixtures of losses**

[4]	$\alpha\mathcal{L}_{\mathrm{Gram}} + \beta\mathcal{L}_{\mathrm{Hist}}$
[5]	$\alpha\mathcal{L}_{\mathrm{Gram}} + \beta\mathcal{L}_{\mathrm{PSD}}$
[6]	$\alpha\mathcal{L}_{\mathrm{Gram}}+\beta\mathcal{L}_{\mathrm{DCorr}}+\gamma\mathcal{L}_{\mathrm{Div}}+\delta\mathcal{L}_{\mathrm{Smooth}}$
[7]	$\alpha\mathcal{L}_{\mathrm{Gram}} + \beta\mathcal{L}_{\mathrm{Disc}} + \gamma\mathcal{L}_{1}$
[8]	$\alpha\mathcal{L}_{\mathrm{Gram}} + \beta\mathcal{L}_{\mathrm{Disc}} + \gamma\mathcal{L}_{1}$

Difficult to use

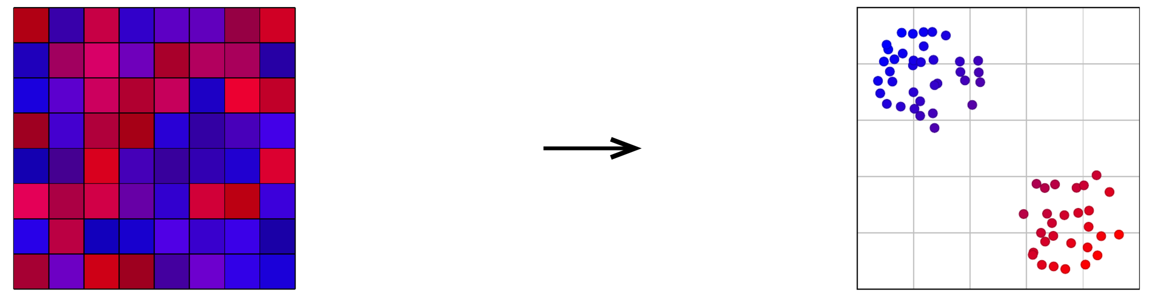
Requires tedious tuning of relative weights.

Compromises convergence.

Conclusion: *No existing loss function is simple to implement, captures relevant statistics completely, and has tractable complexity.*

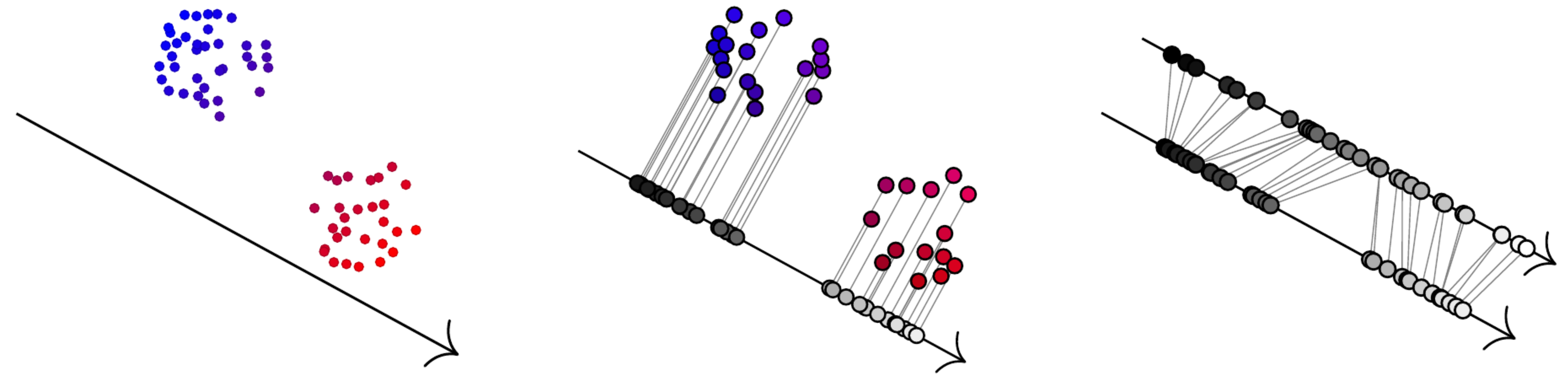
Sliced Wasserstein Loss

We introduce the new loss function $\mathcal{L}_{\mathrm{SW}}$. It does not rely on a subset of feature-activation statistics. Rather, $\mathcal{L}_{\mathrm{SW}}$ evaluates the sliced Wasserstein distance [9] between n-dimensional histograms of feature activations.



H x W n-dimensional features *n-dimensional histogram*

Efficient stochastic histogram distance measurement thanks to *slicing* [9]. The expectation is the nD histogram distance in the optimal transport sense.



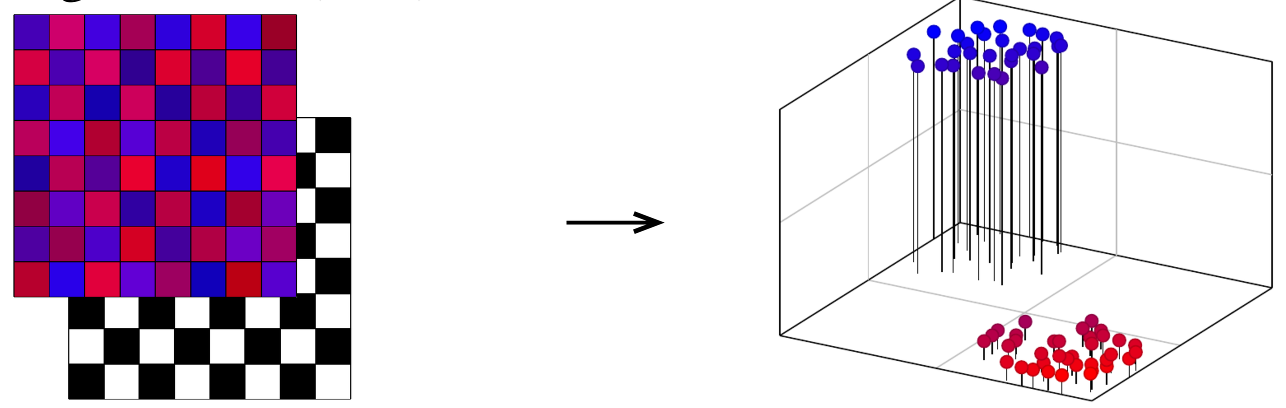
1. Draw K random 1D slices (direction) in nD space

2. Project the nD histogram on each 1D slice

3. Compute L2 distance between (sorted) 1D vectors

Spatial Constraints

$\mathcal{L}_{\mathrm{SW}}$ also supports texture synthesis with spatial constraints. The trick is to concatenate a 1D label or periodicity tag to the feature space. Features of a similar tag are naturally grouped in the (n+1)D histogram: we can proceed using $\mathcal{L}_{\mathrm{SW}}$ in (n+1)D.



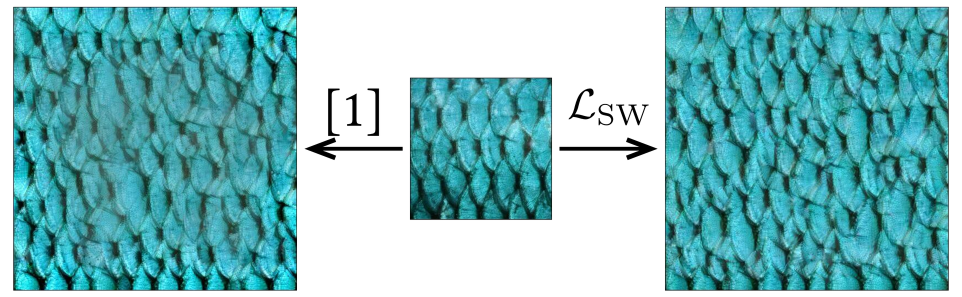
H x W (n+1)-d features *(n+1)-d histogram*

Paper and code available at

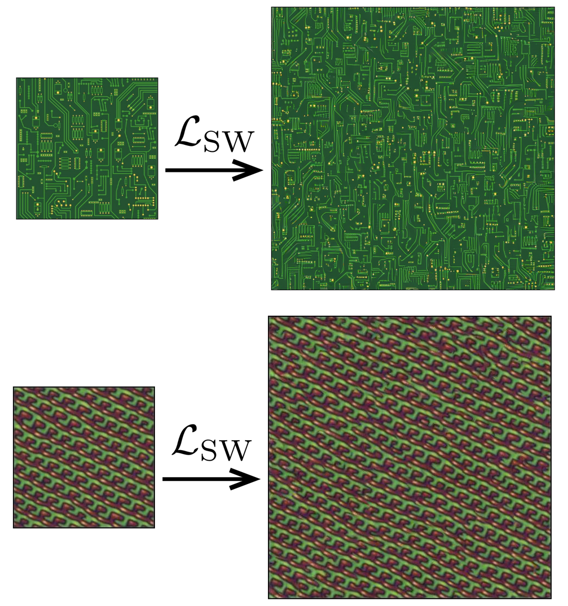
`bit.ly/3wJHNKJ`

Results

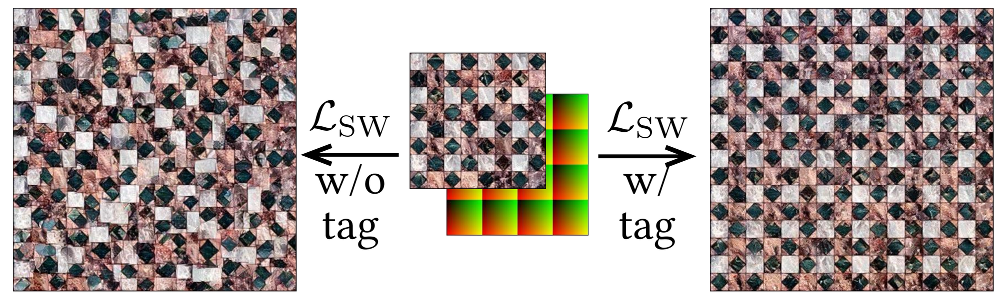
Texture optimization



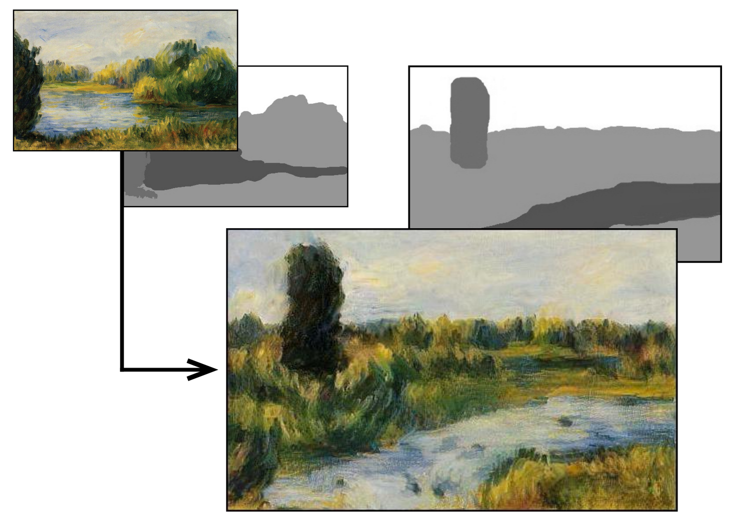
$\mathcal{L}_{\mathrm{SW}}$ improves over Gram



Spatial constraints



Tags preserve periodicity



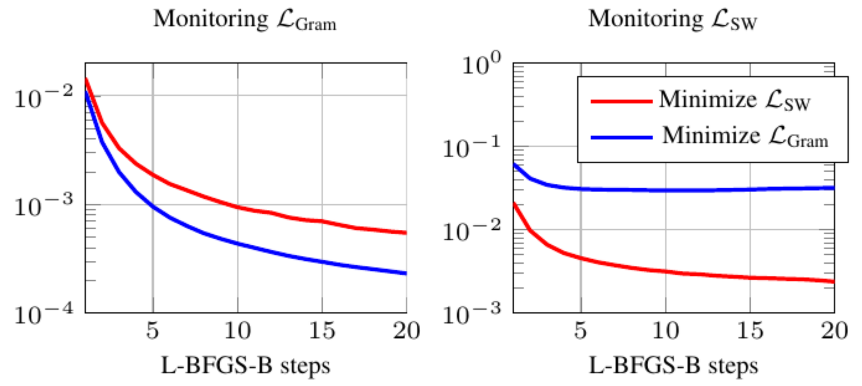
Tags preserve labels

Other results

$\mathcal{L}_{\mathrm{SW}}$ is 1.7 - 2.8x more costly than Gram

$\mathcal{L}_{\mathrm{SW}}$ allows to train a multi-texture generator

$\mathcal{L}_{\mathrm{SW}}$ encompasses $\mathcal{L}_{\mathrm{Gram}}$: minimizing it reduces $\mathcal{L}_{\mathrm{Gram}}$. The opposite is false.



[1] Texture synthesis using convolutional neural networks, Gatys et al., NIPS 2015
[2] Style transfer by relaxed optimal transport and self-similarity, Kolkin et al., CVPR 2019
[3] The contextual loss for image transformation with non-aligned data, Mechrez et al., ECCV 2018
[4] Stable and controllable neural texture synthesis and style transfer using histogram losses, Risser et al., Arxiv 2017
[5] Texture synthesis through convolutional neural networks and spectrum constraints, Liu et al., ICPR 2016
[6] Deep correlations for texture synthesis, Sendik et al., SIGGRAPH 2017
[7] Non-stationary texture synthesis by adversarial expansion, Zhou et al., SIGGRAPH 2018
[8] Texture mixer: a network for controllable synthesis and interpolation of texture, Yu et al., CVPR 2019
[9] N-dimensional probability density function transfer and its applications to color transfer, Pitie et al., ICCV 2005

All our results are produced using our single loss term.