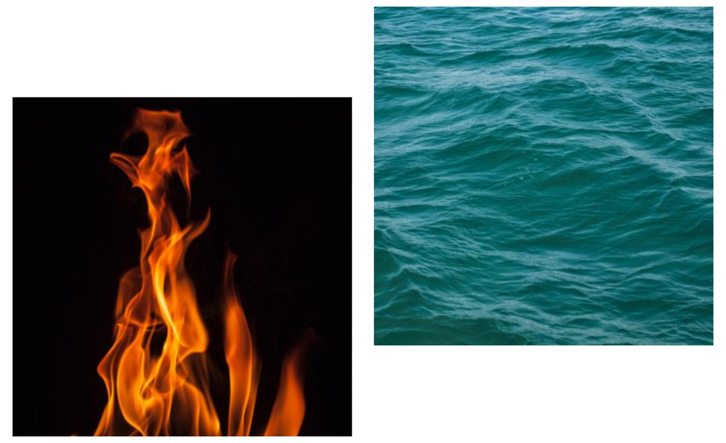
DyNCA: Real-Time Dynamic Texture Synthesis Using Neural Cellular Automata

Ehsan Pajouheshgar*, Yitao Xu*, Tong Zhang, Sabine Süsstrunk

EPFL · IVRL

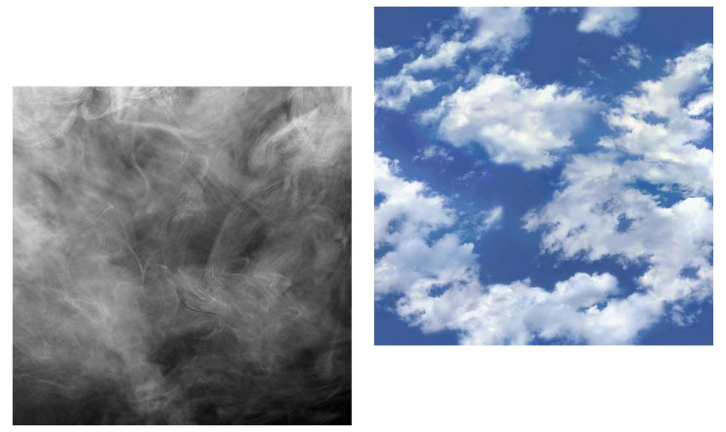
JUNE 18-22, 2023 CVPR VANCOUVER, CANADA

Problem



Textures are everywhere. We perceive them as spatially repetitive patterns. Dynamic Textures are textures that change over time inducing a sense of motion.

The goal of Dynamic Texture Synthesis is to generate perceptually-equivalent videos of an exemplar dynamic texture.



Contributions

dynca.github.io

Link to our Demo

Our method can synthesize Dynamic Texture Videos in **real-time** achieving **10^2~10^4 speedup** compared to the SOTA methods.

Method	I	II	III	IV	V	VI	VII
Doretto et al [2003]	✗	✓	✓	✗	✓	✗	✗
Costantini et al. [2007]	✗	✓	✓	✗	✓	✗	✗
Funke et al. [2017]	✗	✓	✓	✗	✗	✗	✗
Xie et al. [2017]	✗	✗	✗	✗	✓	✗	✗
Tesfaldet et al. [2018]	✗	✓	✗	✗	✗	✓	✓
Zhang et al. [2021]	✗	✓	✗	✗	✗	✓	✓
DyNCA (Ours)	✓	✓	✓	✓	✗	✓	✓

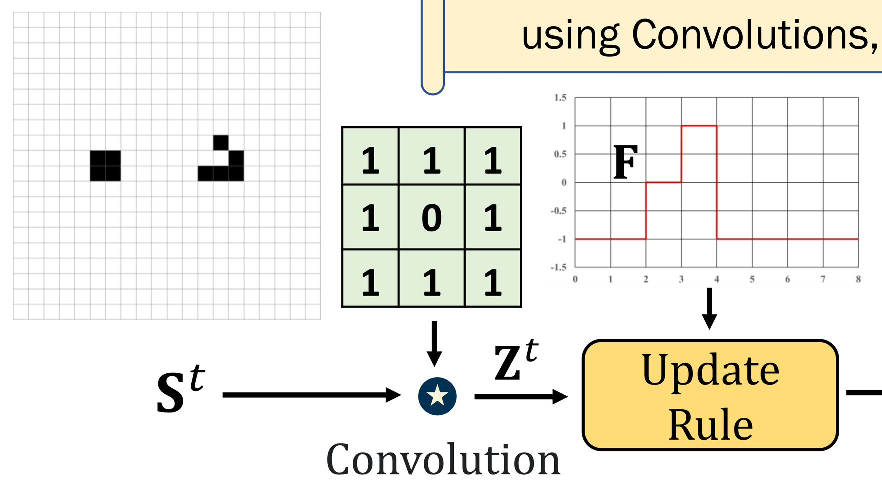
(I) Arbitrary Resolution
(II) Arbitrary long videos
(III) Synthesize new samples
(IV) Real-time video editing
(V) Require pretrained models
(VI) Disentangled appearance and motion.
(VII) Vector field supervision

Cellular Automata and Conway's Game of Life

Grid: State of the cell at location i, j at time t is S_{ij}^t
Neighborhood: Cells can perceive their neighbors
Update rule: How cell states change at each step
Starting condition: Initial state of the cells, i.e., S_{ij}^0

Game of Life
- Survive if 2 or 3 neighbors are alive
- Become alive if 3 neighbors are alive
- Die or remain dead

Cellular Automata can be implemented using Convolutions, Gilpin [2019].



Perception
$$Z^t = K * S^t$$

Update Rule
$$\Delta S_{ij}^t = F(Z_{ij}^t)$$
$$S^{t+1} = S^t + \Delta S^t$$

Architecture

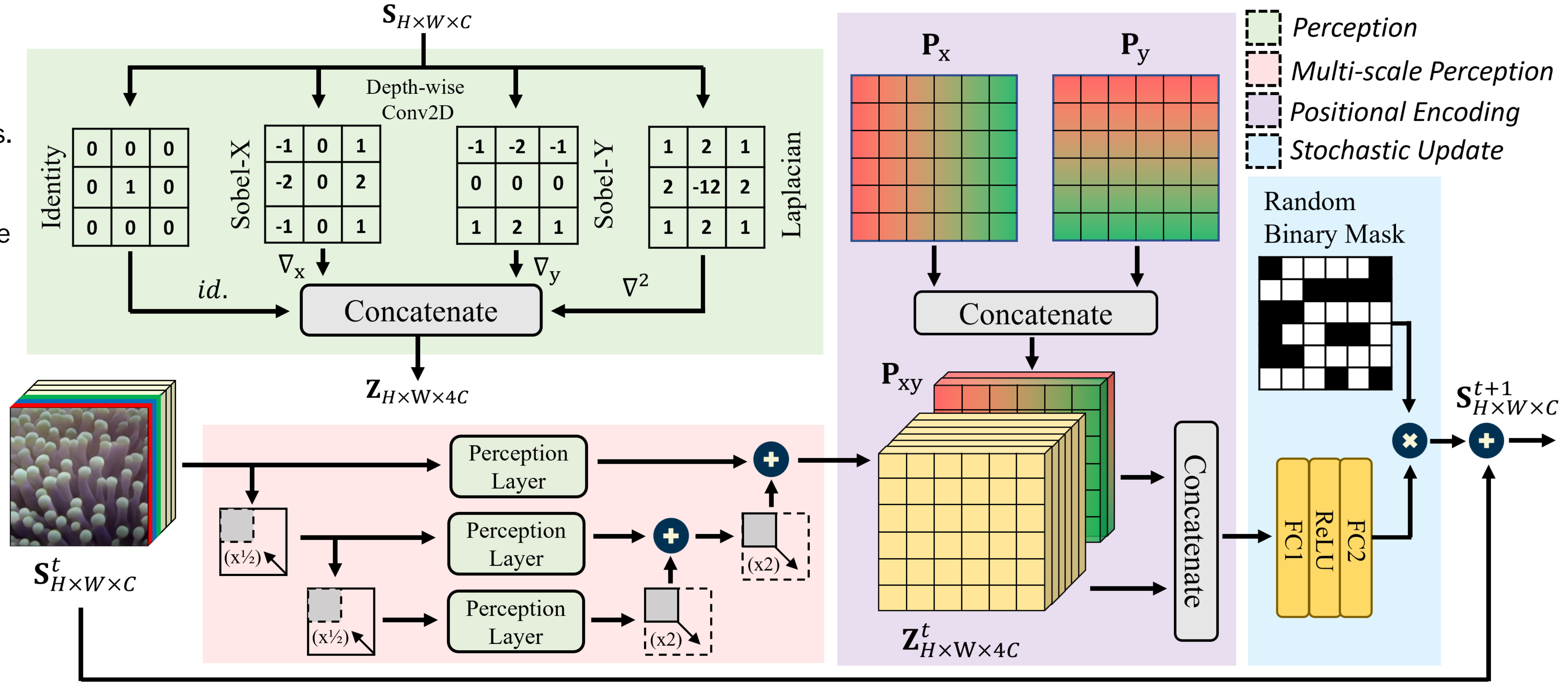
❑ *Cell States*
State is a C dimensional vector. The first 3 dimensions are the RGB values.

❑ *Perception*
Four fixed convolution kernels that are frozen during the model's training.

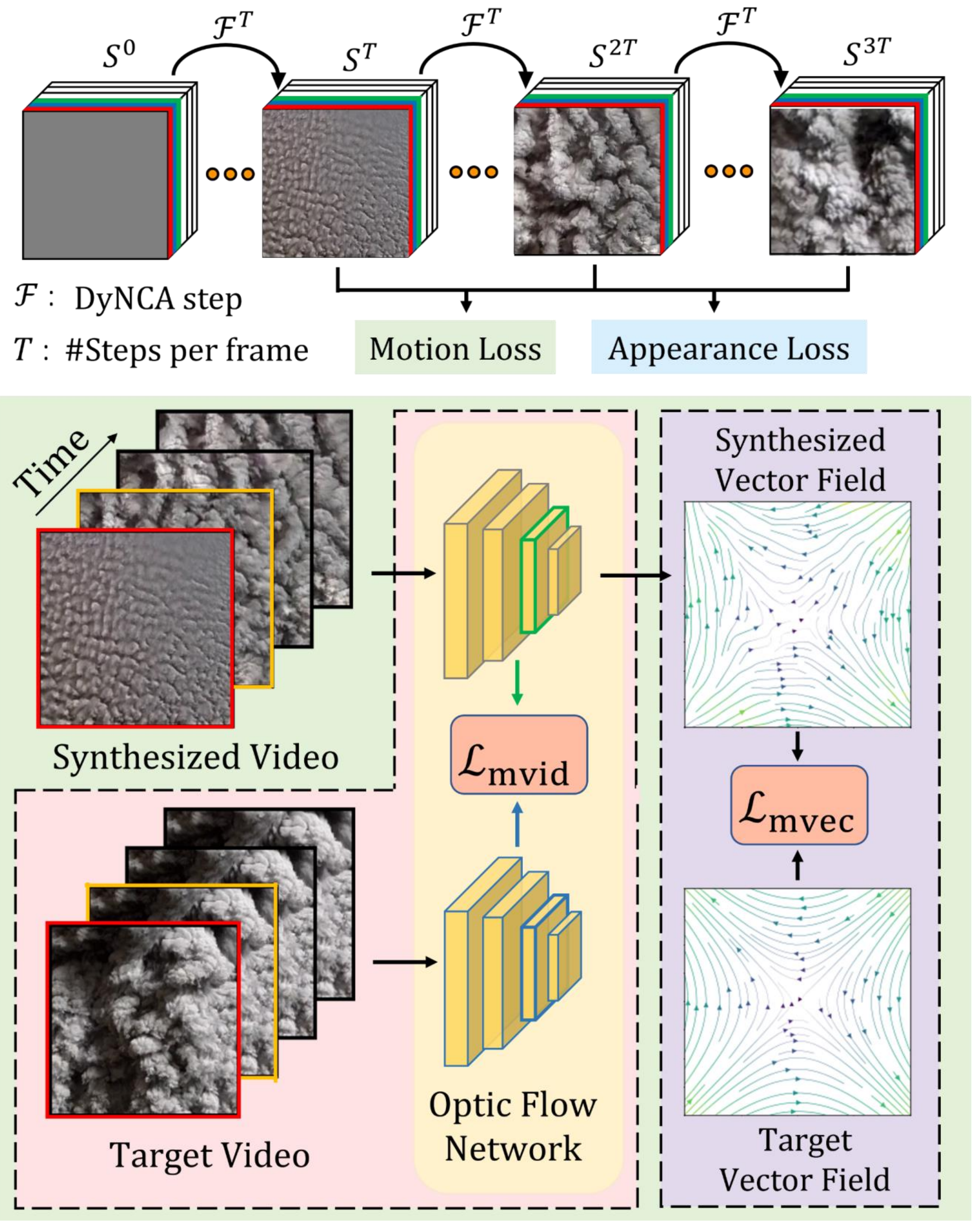
❑ *Multi-scale Perception*
Increase the communication range of the cells and improve stability.

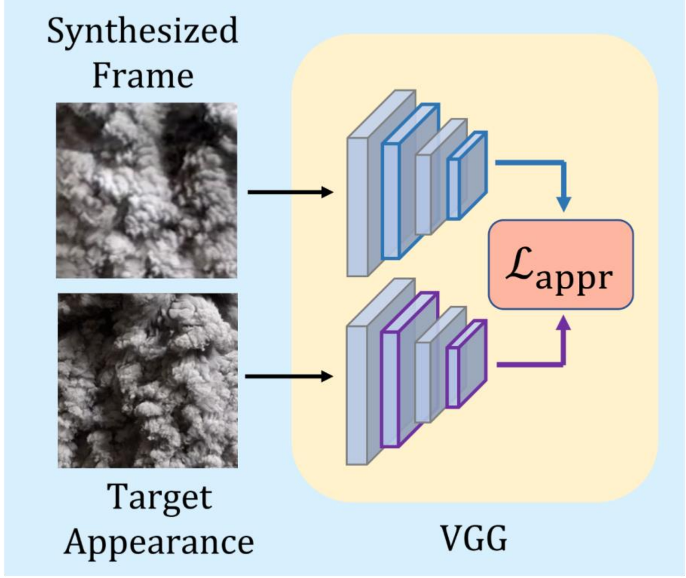
❑ *Positional Encoding*
Allows the cells to be aware of their global position in the grid.

❑ *Stochastic Update*
The *update rule* is represented by two trainable FC layers and a random binary update mask.



Training and Loss Functions





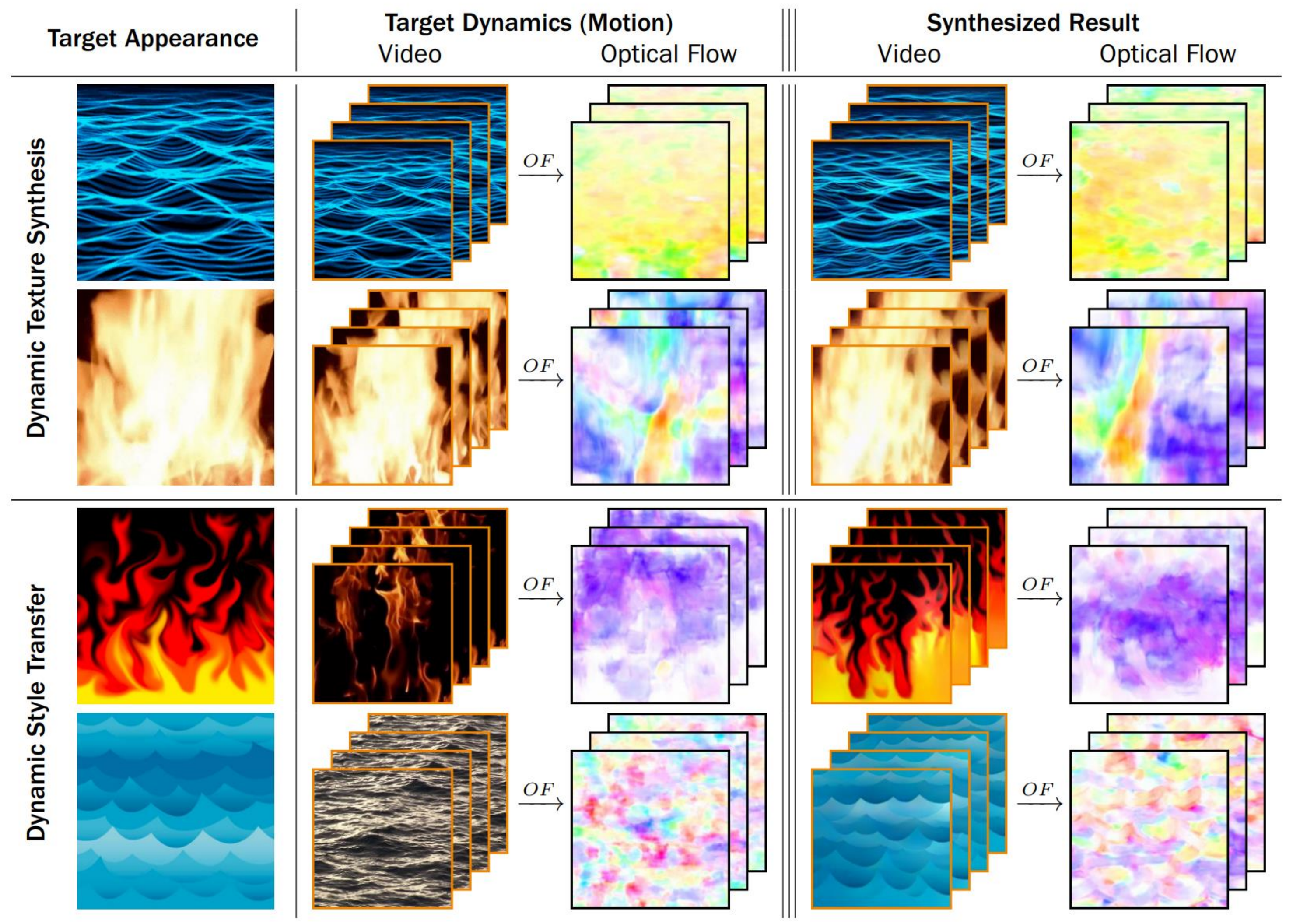
Style Matching Loss (Kolkin et al. [2019])

$$\mathcal{L}_{appr} = \frac{1}{K}\sum_{k=1}^{K}\sum_{l=1}^{5}\left(\mathcal{L}_s(X_k^l, Y^l) + \mathcal{L}_m(X_k^l, Y^l)\right)$$

$$\mathcal{L}_{mvid} = \frac{1}{K-1}\sum_{k=1}^{K-1}\left(\mathcal{L}_s(X_k, Y_k), \mathcal{L}_m(X_k, Y_k)\right)$$

$$\mathcal{L}_{mvec} = (1.0 - \min\{1.0, \mathcal{L}_{dir}\})\mathcal{L}_{norm} + \gamma\mathcal{L}_{dir}$$

Results



Top: DyNCA captures the appearance and the motion from a video. Bottom: DyNCA also disentangles the appearance and motion and performs Dynamic Style Transfer when the target dynamics are different.

Comparisons and User Study

Method	Res.	Synthesis Time	# Parameters
A	256^2	500s	0.2M/frame
B	224^2	400s	81M
C	100^2	8.5s	2.8M
DyNCA-S	128^2	0.033s	0.006M
DyNCA-S	256^2	0.057s	0.006M
DyNCA-L	128^2	0.035s	0.01M
DyNCA-L	256^2	0.057s	0.01M

	Real	DyNCA	A	B	C
Real	N/A	27%	26%	24%	8%
DyNCA	73%	N/A	40%	46%	20%
A	74%	60%	N/A	52%	25%
B	76%	54%	48%	N/A	15%
C	92%	80%	75%	85%	N/A

We show videos to the participants and ask them to chose the video that appears the most realistic. (A) Tesfaldet et al. [2018]. (B), and (C) two different configurations from Xie et al. [2017].