

It is real?

Improving simulations with a deep learning discriminator



Bartłomiej Olechno

Piotr Migdał



ML in PL
Virtual Event 2020

18 Dec 2020

<https://virtual-event.mlinpl.org/>

Goal

Realistic simulations in racing games

- AI for physics engine optimization
- Assisting gamedev for realism of drifting

Outline

1. Deep learning for video games
2. GAN-like architecture
3. Predict/update pipeline
4. What's next?



Bartłomiej Olechno



Piotr Migdał

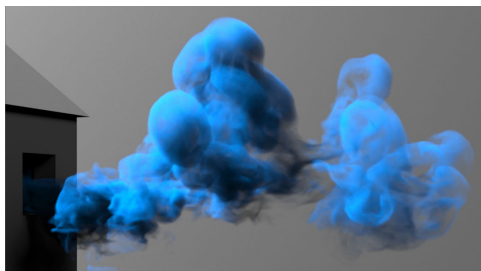
Deep learning for realistic animations and physics



Higher detail and performance for ray-tracing on GPUs.

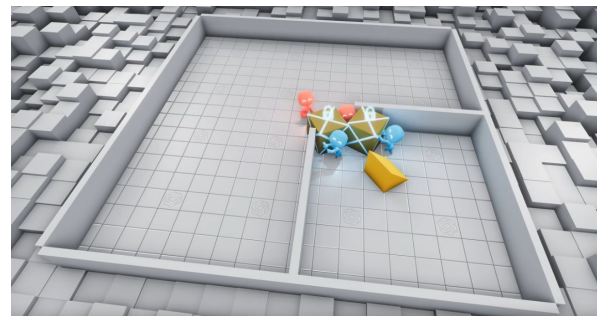
Used in Cyberpunk 2077.

[DLSS 2.0 by Nvidia](#) (2020)



Faster smoke, flame and water simulation.

[tempoGAN by You Xie et al.](#) (2018)



Agents searching for simulator and environment exploits.

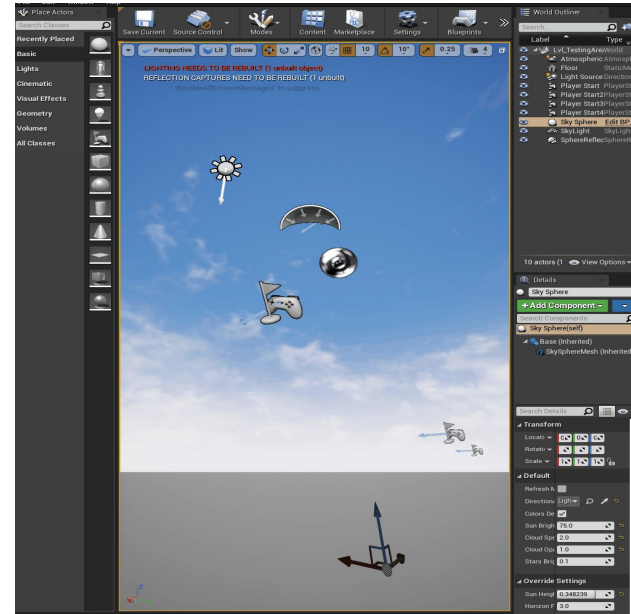
[Emergent Tool Use by OpenAI](#) (2019)

Data source: measured & generated time series

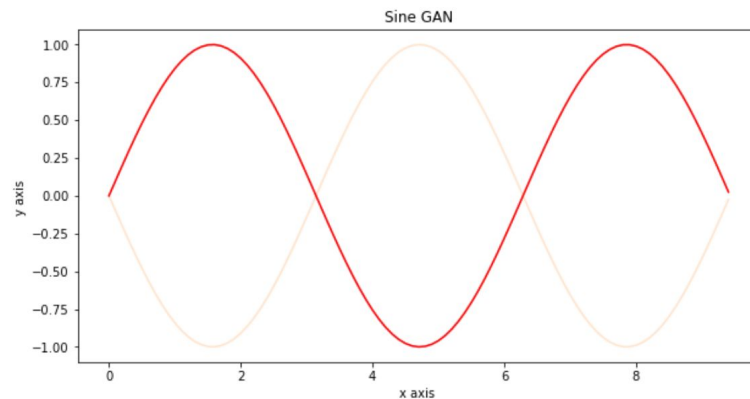
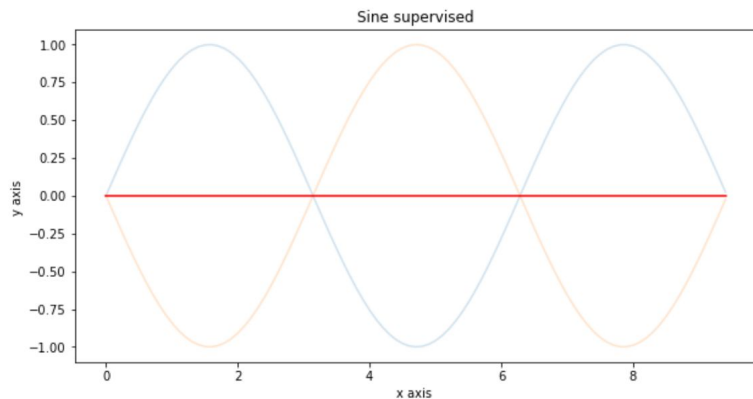
Real race car telemetry
(experiments in Gdańsk)



Computer simulations
(in Unity3d and Unreal Engine)



Why not (self)supervised anomaly detection?



Supervised methods promote ‘averaged’ guess

In the case of super resolution GAN supervised technique gives blurry result.

[SRGAN Ledig et al. \(2016\)](#)



Simulating a race car

(HYPER)PARAMETERS

INPUT SEQUENCE

Brake Pos	Clutch Pos	Gear	Steering Angle	Throttle Pos
0.000000	0.000000	3	-0.542500	1.000000
0.000000	0.000000	3	-1.127000	1.000000
0.000000	0.000000	3	-1.708000	1.000000
0.000000	0.000000	3	-2.292500	1.000000
0.000000	0.000000	3	-2.957500	1.000000
0.000000	0.000000	3	-3.587500	1.000000
0.000000	0.000000	3	-4.172000	1.000000
0.000000	0.000000	3	-4.756500	1.000000
0.000000	0.000000	3	-5.341000	1.000000
0.000000	0.000000	3	-5.971000	1.000000
0.000000	0.000000	3	-6.562500	1.000000



Black box
game engine
(no assumptions on
differentiability)



OUTPUT SEQUENCE

Velocity Y	Velocity X	Velocity Z	Chassis Pitch Angle
-0.522769	22.749952	-2.769650	-1.333294
-0.483330	22.746174	-2.762518	-1.110568
-0.444166	22.741568	-2.754680	-0.887426
-0.405128	22.736673	-2.746430	-0.663758
-0.361648	22.748146	-2.739610	-0.406048
-0.320994	22.732609	-2.729396	-0.162176
-0.283872	22.714958	-2.719617	0.060932
-0.244858	22.708906	-2.711789	0.284901
-0.206119	22.702196	-2.703089	0.509307
-0.163956	22.704609	-2.695145	0.754674
-0.126793	22.686518	-2.684933	0.981252
-0.080899	22.700447	-2.677609	1.257595
-0.049809	22.666138	-2.666056	1.479241
-0.008225	22.676062	-2.625984	1.737791
0.036085	22.670858	-2.579572	2.000118
0.103681	22.617065	-2.817481	2.163196
0.151375	22.586151	-2.996169	2.273401

Does it simulate REALISTIC physics of the SELECTED car?

Discriminator for real vs simulated (a GAN-like approach)

INPUT + OUTPUT
SEQUENCE

Steering Angle	Throttle Pos	Velocity Y	Velocity X
-0.542500	1.000000	-0.522769	22.749952
-1.127000	1.000000	-0.483330	22.746174
-1.708000	1.000000	-0.444166	22.741568
-2.292500	1.000000	-0.405128	22.736673
-2.957500	1.000000	-0.361648	22.748146
-3.587500	1.000000	-0.320994	22.732609
-4.172000	1.000000	-0.283872	22.714958
-4.756500	1.000000	-0.244858	22.708906
-5.341000	1.000000	-0.206119	22.702196



Discriminator
(neural network)



72% REAL

28% SIMULATED

A better approach: Fully Convolutional NN, so we get “realism” for each timeframe

INPUT + OUTPUT
SEQUENCE

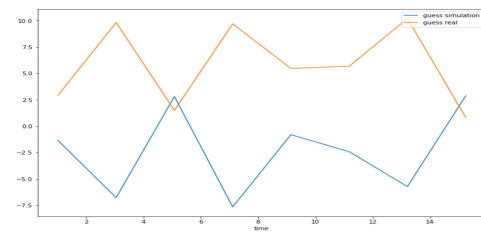
Steering Angle	Throttle Pos	Velocity Y	Velocity X
-0.542500	1.000000	-0.522769	22.749952
-1.127000	1.000000	-0.483330	22.746174
-1.708000	1.000000	-0.444166	22.741568
-2.292500	1.000000	-0.405128	22.736673
-2.957500	1.000000	-0.361648	22.748146
-3.587500	1.000000	-0.320994	22.732609
-4.172000	1.000000	-0.283872	22.714958
-4.756500	1.000000	-0.244858	22.708906
-5.341000	1.000000	-0.206119	22.702196



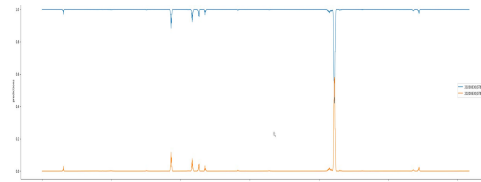
Discriminator
(Fully-CNN neural
network)



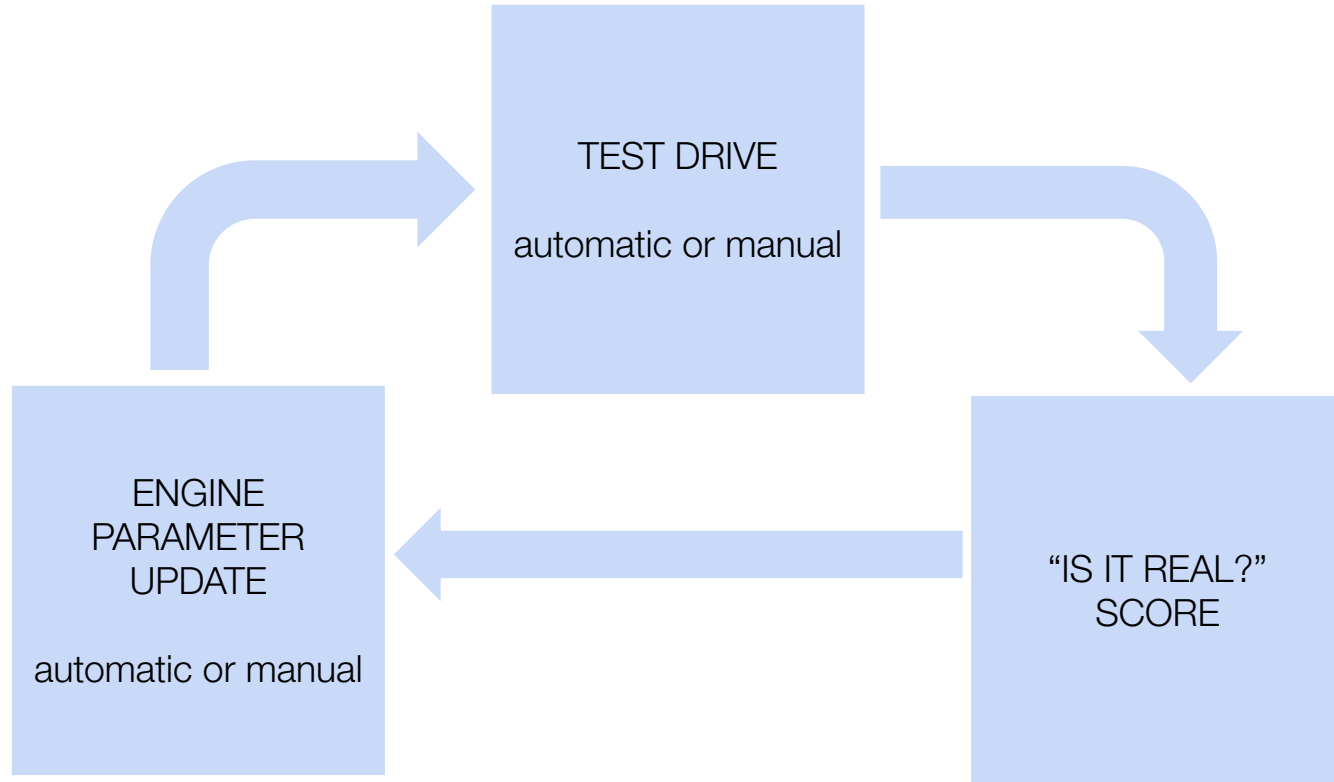
REAL/SIMULATED



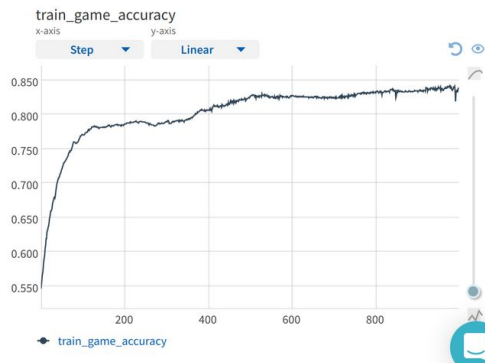
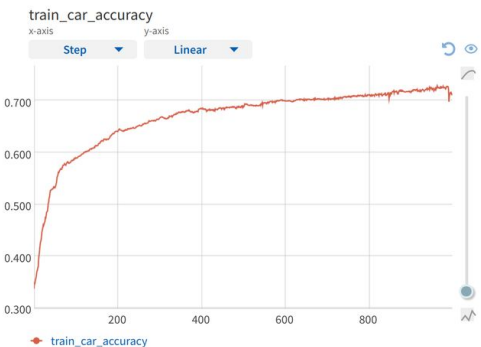
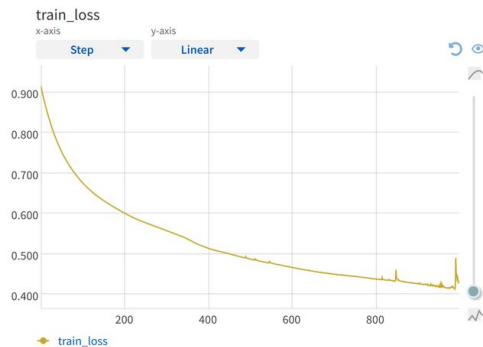
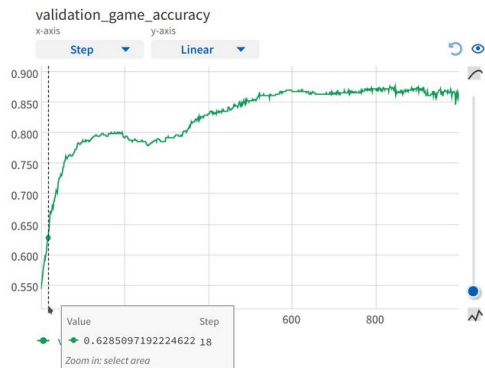
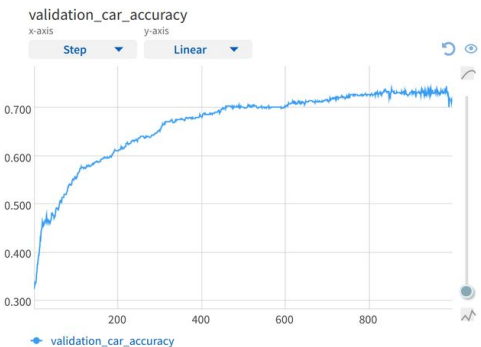
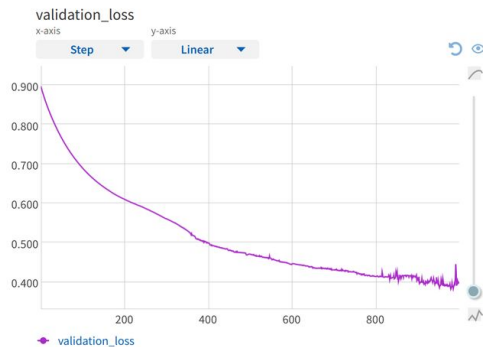
CAR TYPE



An iterative approach to improve simulation



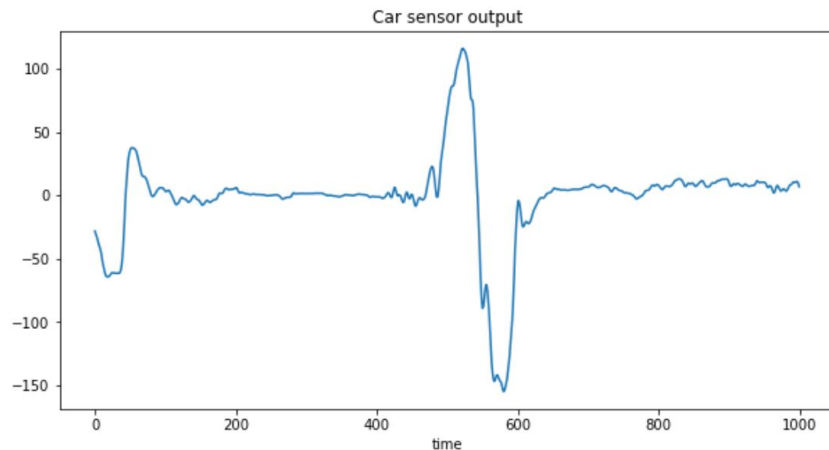
Experiments management and visualization



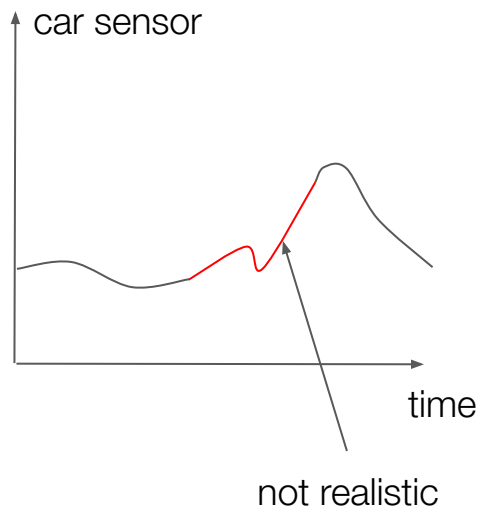
[livelossplot](#) and [neptune.ml](#)

Challenges: not all artifacts are equal

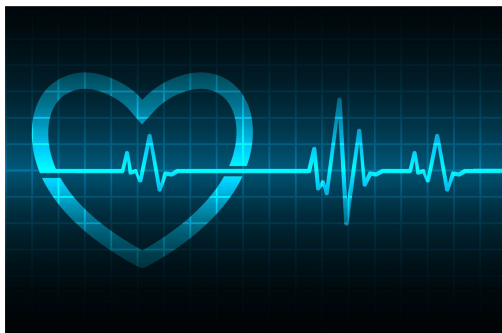
- CNNs are oversensitive to local patterns
- Some local patterns (e.g. noise) are easy for discrimination, but do not change the long-term car behavior oraz general experience



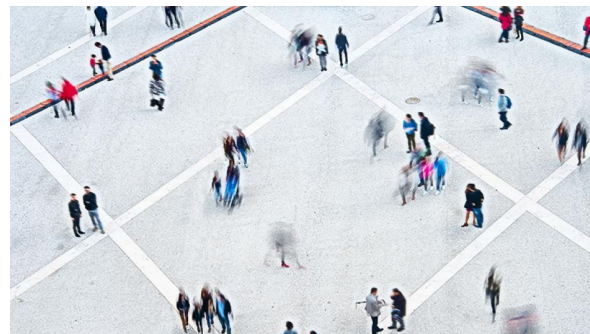
Further use cases



Testing realism of car physics.



Testing realism of medical simulations (e.g. ECG, heart blood flow).

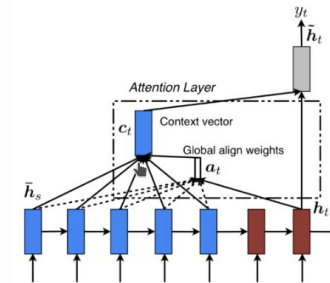


Testing realism of body movement and crowd behavior (for animations, games, simulations in sociology.)

Bibliography

- [M. Chu, N. Thuerey Data-Driven Synthesis of Smoke Flows with CNN-based Feature Descriptors \(2017\)](#)
- [Kim et al. Deep Fluids: A Generative Network for Parameterized Fluid Simulations \(2018\)](#)
- [Werhahn et al. A Multi-Pass GAN for Fluid Flow Super-Resolution \(2019\)](#)
- [Xie et al. tempoGAN: A Temporally Coherent, Volumetric GAN for Super-resolution Fluid Flow \(2018\)](#)
- [Baker et al. Emergent Tool Use From Multi-Agent Interaction \(2019\)](#)
- [Giulio Jiang and Bernhard Kainz One Shot Radiance: Global Illumination Using Convolutional Autoencoders \(2019\)](#)
- [Ledig et al. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network \(2016\)](#)
- [Nvidia DLSS](#)

Thank you for your attention!



If you have a question please contact me at:

bolechno@eccgames.com

This project is supported by a European program *Program Operacyjny Inteligentny Rozwój* for [GearShift - building the engine of behavior of wheeled motor vehicles and map's generation based on artificial intelligence algorithms implemented on the Unreal Engine platform](#) lead by ECC Games (NCBR grant GameINN).