Learning Visual Predictive Models of Physics

Tom Sanitz

25. February 2024

Table of Contents

Motivation

Motivation

- Humans can predict the motion of objects
- We do not solve equations of motion
- Imagination of trajectory
- Like running an internal 'simulation'

How to acquire this imagination?

Visual Imagination

- Knowledge of both agent and world required
- Modeling the external world very complex
- Learning imagined trajectory from visual input alone?

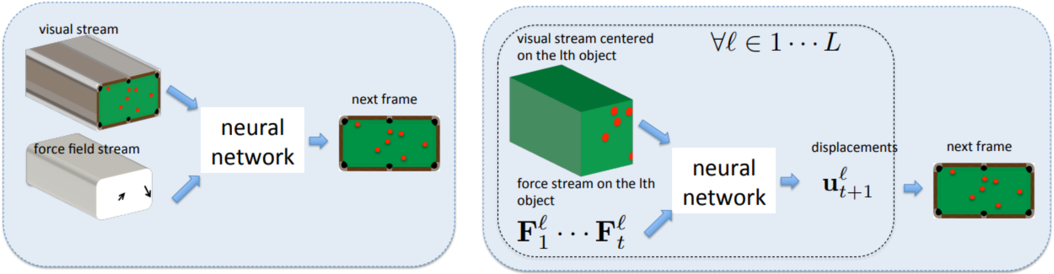
Learning Visual Predictive Models

- 'Learning visual predictive models of physics for playing billiards' by Katerina Fragkiadaki, Pulkit Agrawal, Sergey Levine, Jitendra Malik [1]

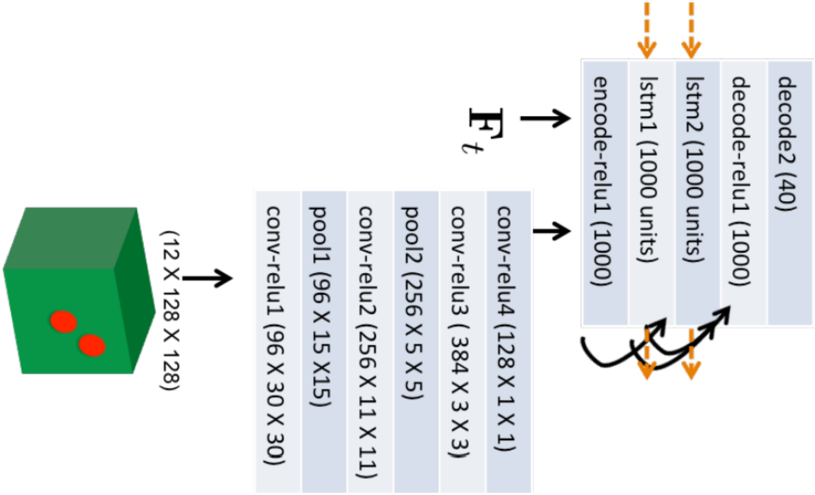
Object-Centric Prediction

Object-Centric Benefits

- Naturally includes translation invariance
- Easily share model across different 'worlds'



Network Architecture



Network Architecture

Input at each time step

1. Current + previous 3 glimpses (images)
2. Applied forces $F_t = (F_t^x, F_t^y)$
3. Hidden states of LSTM units $t - 1$

Network output

- Ball displacement $u_{t+k} = (\lambda x_{t+k}, \lambda y_{t+k})$ for $k = 1 \dots h$ in next h frames
- Predict next 20 steps, therefor $20 \times 2 = 40$ output values

Model Training: World Setup

Random configurations:

- Rectangular and non-rectangular walls
- Wall length[300 pixel, 550 pixel]
- Starting point
- Forces on the ball (first frame only)
- Sequence length([20,200]

Model Training: Loss

- Weighted Euclidean Loss
- Errors in shorter time horizon get higher loss

Loss Function

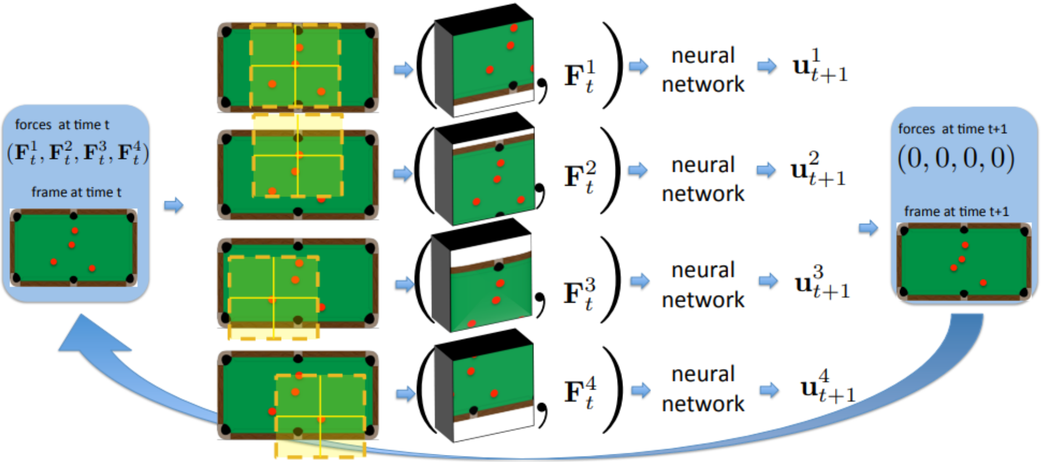
$L = \sum_{k=1}^{h} w_k ||\widetilde{u}_{t+k} - u_{t+k}||_2^2$

Visual Imagination

Generate Visual Imaginations

- Predicted trajectory leads to generate visual imaginations?
- Translate each ball by predicted velocity ($\widetilde{u}_t$) at time t
- Repeat iteratively for all future world states

Evaluation: Imagination



Model Evaluation

Error in angle and magnitude

- Constant velocity (CV)
- Object centric (OC)
- Compared to frame centric (FC)

Model Evaluation

Evaluation Rectangular World

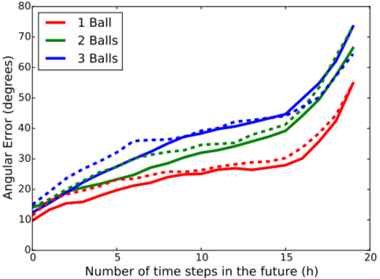
- Near Collision := within [-4,4] frames depicting collision
- Mean angular error in degrees
- Relative error in magnitude of predicted velocity

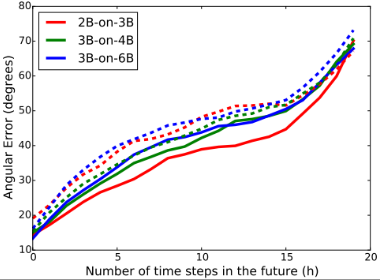
Time	Overall Error			Error Near Collisions		
	CV	FC	OC	CV	FC	OC
t+1	$3.0^o/0.00$	$6.2^o/0.04$	$5.1^o/0.03$	$23.2^o/0.00$	$11.4^o/0.06$	$9.8^o/0.04$
t+5	$11.8^o/0.01$	$8.7^o/0.05$	$7.2^o/0.04$	$56.6^o/0.05$	$21.1^o/0.12$	$17.9^o/0.10$
t+20	$45.3^o/0.01$	$16.3^o/0.09$	$14.8^o/0.09$	$123.0^o/0.04$	$54.8^o/0.20$	$54.8^o/0.20$

Evaluation: Object Centric vs Frame Centric

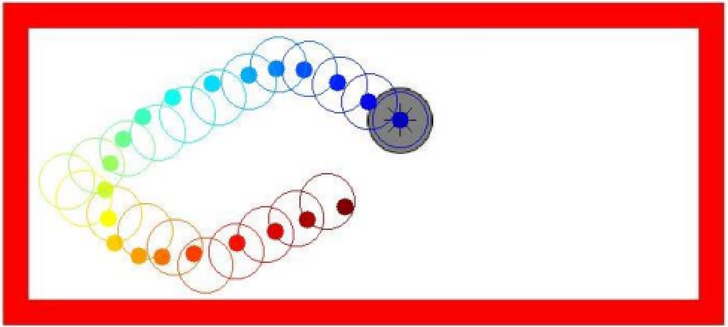
Comparison Details

- Near collision angular error
- Dashed := FC, solid := OC
- 20 steps (h=20)
- 2B-on-3B := trained on 2 ball world, eval on 3 ball world

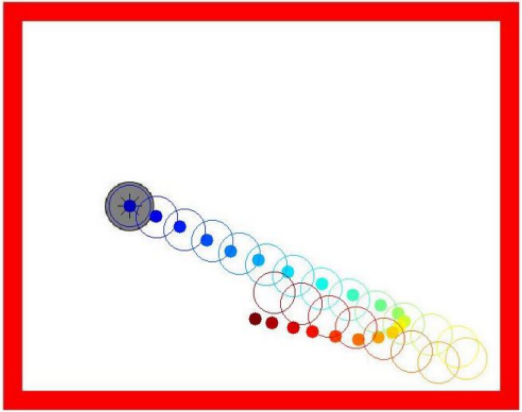




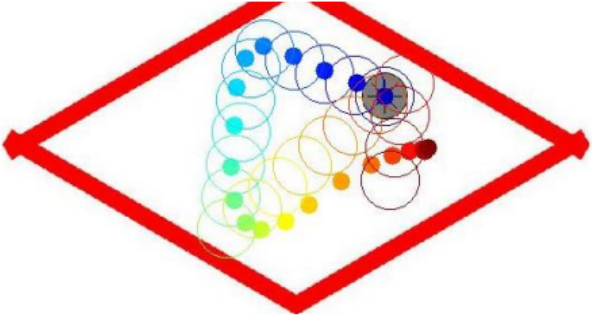
Qualitative Evaluation



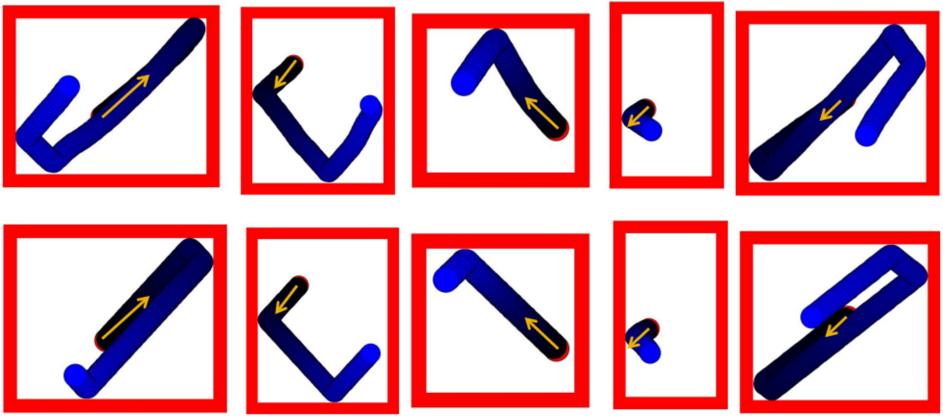
Qualitative Evaluation



Qualitative Evaluation



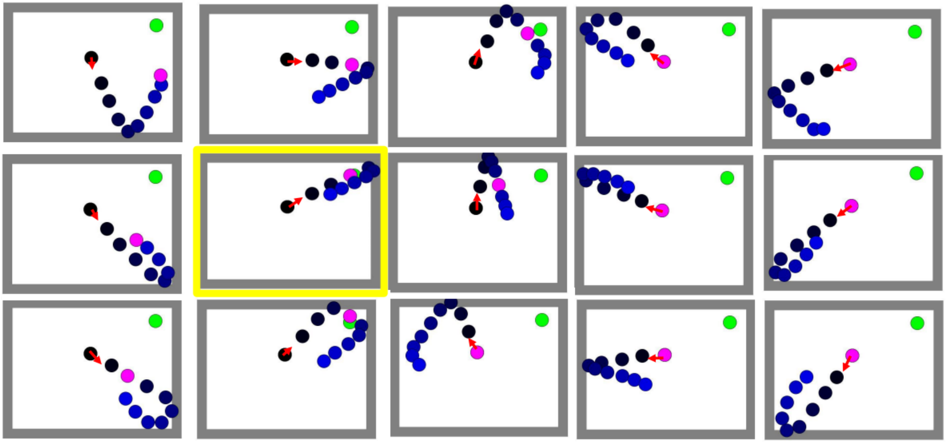
Evaluation: Visual Imagination



Action Planning Using Visual Predictions

- Plan actions for which the agent was never trained
- Planning force required to push ball to desired location
- Achieved using:
  1. Run multiple visual imaginations (simulations)
  2. Optimal force = Closest ball to target location

Action Planning Using Visual Predictions



Results: Action Planning Using Visual Predictions

- OC-Model outperforms FC-Model
- Oracle is the physics simulator
- Hit accuracy in amount of tries, where ball in required distance to target
- Arena size: 300-550 pixel

Method	Hit Accuracy		
	< 10 pixels	< 25 pixels	< 50 pixels
Oracle	95%	100%	100%
Random	3%	14%	23%
Ours (FC-Model)	15%	39%	60%
Ours (OC-Model)	30%	56%	85%

Similarities And Challenges For The Project

- Top down view and 2D trajectories very similar to our golf ball
- Initially planned to use a similar approach, but long term errors are accumulating
- Most likely improvement using Transformers?
- Overall probably inferior to learning a residual like in Tossingbot [2]
- However could be considered for local patches, e.g. infront of obstacles

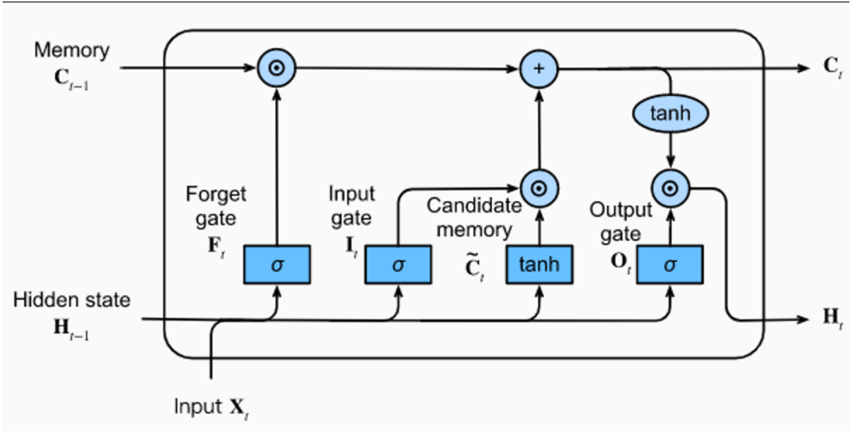
Thank you for your attention!

References

[1] Katerina Fragkiadaki, Pulkit Agrawal, Sergey Levine, and Jitendra Malik. "Learning visual predictive models of physics for playing billiards". In: arXiv preprint arXiv:1511.07404 (2015).

[2] Andy Zeng, Shuran Song, Johnny Lee, Alberto Rodriguez, and Thomas Funkhouser. "Tossingbot: Learning to throw arbitrary objects with residual physics". In: IEEE Transactions on Robotics (2020).

Backup: LSTM


Memory
$\mathbf{C}_{t-1}$
$\mathbf{C}_t$
tanh
+
Forget
gate
$\mathbf{F}_t$
σ
Input
gate
$\mathbf{I}_t$
σ
Candidate
memory
$\tilde{\mathbf{C}}_t$
tanh
Output
gate
$\mathbf{O}_t$
σ
Hidden state
$\mathbf{H}_{t-1}$
$\mathbf{H}_t$
Input $\mathbf{X}_t$