Unsupervised Learning of Object Keypoints for Perception and Control

Presenter: Aryaman Singh Samyal

21st of September 2021

Problem Setting

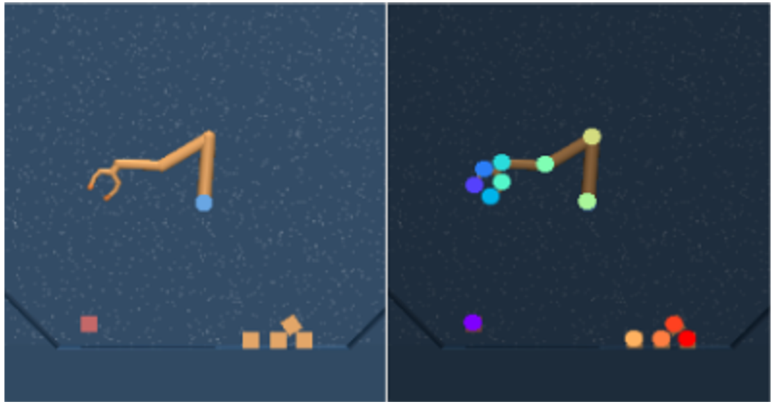
- Object representation in computer vision is primarily focused on image classification, object detection or segmentation.
- Not very useful for reinforcement learning tasks involving control of an agent.
- For reinforcement learning and control you require precise spatial geometric representation of the object.
- Also, feature representation are usually task-specific which makes it difficult to re-purpose the learnt knowledge to unseen domains/objectives

Motivation - How’s it important?

- Object representations are vital for robots to perceive the world around them.
- Precise geometric representations of objects would enable robots to make better decisions.
- Understanding of keypoints of an object are critical in control and RL domains.
- Improvement in this domain directly impacts and closes the gap between sim2real transfer for robotics.

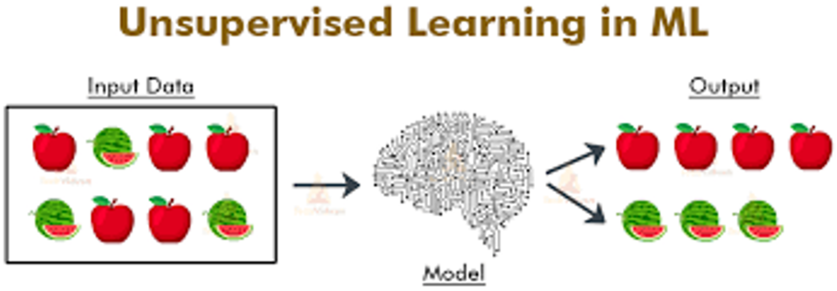
Research Objectives

- To find object representations in that are useful for control and reinforcement learning.
- Discover keypoints of an object in a task-agnostic manner for accurate geometric representation.
- Track the keypoints consistently and accurately over long-time horizons or multiple frames.



Unsupervised Learning

- Algorithm is not provided with any pre-labeled data for training.
- During training will sort the data according to similarity of features and learning patterns within the dataset.
- Utilized for applications with an abstract objective/target.



Q-Learning

- Off-policy reinforcement learning algorithm to learn the value of an action given the current state.
- Learns a policy that maximizes the total reward.
- Provides a balance between exploring and exploiting.

```
import numpy as np

# Initialize q-table values to 0

Q = np.zeros((state_size, action_size))
```

Q-Learning

```
import random

# Set the percent you want to explore
epsilon = 0.2

if random.uniform(0, 1) < epsilon:
    """
    Explore: select a random action

    """
else:
    """
    Exploit: select the action with max value (future reward)

    """
```

Q-Learning

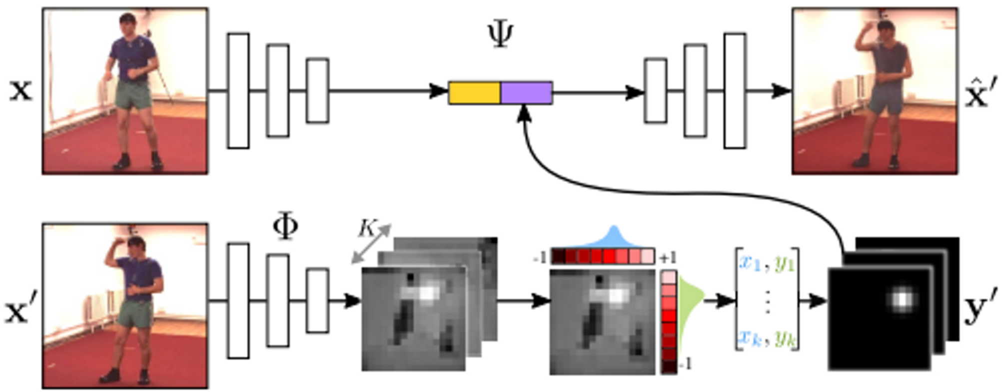
```
# Update q values

Q[state, action] = Q[state, action] + lr * (reward + gamma *
np.max(Q[new_state, :]) — Q[state, action])
```

Related work: Conditional image generation

- In their paper, Jakab and Gupta et al. propose an encode-decoder architecture to extract keypoints by introducing a bottleneck to distil geometry related features without supervision.
- Keypoints are extracted from two example pictures with an object with different viewpoint or deformation.
- Bottleneck architecture is applied by the authors for this paper's proposed method.
- **Limitations**
  - Not as consistent over long-term tracking
  - Can learn non-spatial latent embedding.

Related work: Conditional image generation

Model architecture as proposed by Jakab and Gupta et al.



Related work: Autoencoder for keypoint discovery

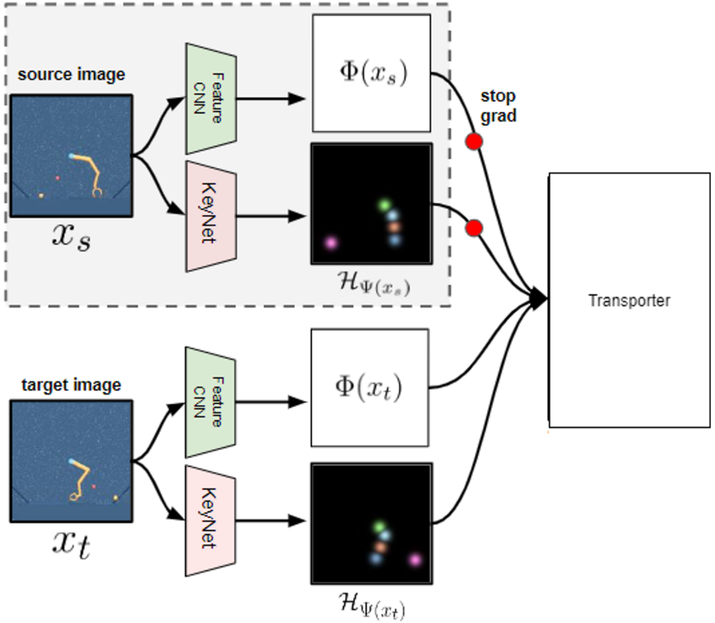
- Zhang et al. in their paper propose an unsupervised, autoencoding method to discover keypoints or “landmarks” from a single-image.
- Outputs semantically meaningful keypoint coordinates as an explicit structure representation.
- **Limitations**
  - Requires temporal information between frames in the form of optical flow.
  - Multiple loss and regularization terms for convergence.
  - Poor long-term performance in tracking keypoints over frames.

Proposed Method: Transporter

- Authors propose "Transporter", a neural network architecture to detect keypoints for object representations that useful for the control and RL domain.
- Utilizes unsupervised learning and learns using a source and a target frame, given only raw videos.
- The keypoints generated, consistently and accurately track the object as it undergoes deformation.
- **Goal** - Extract K number of 2D locations or "keypoints" which correspond to object or moving entities of an object without any manual labels.

Keypoint and feature extraction

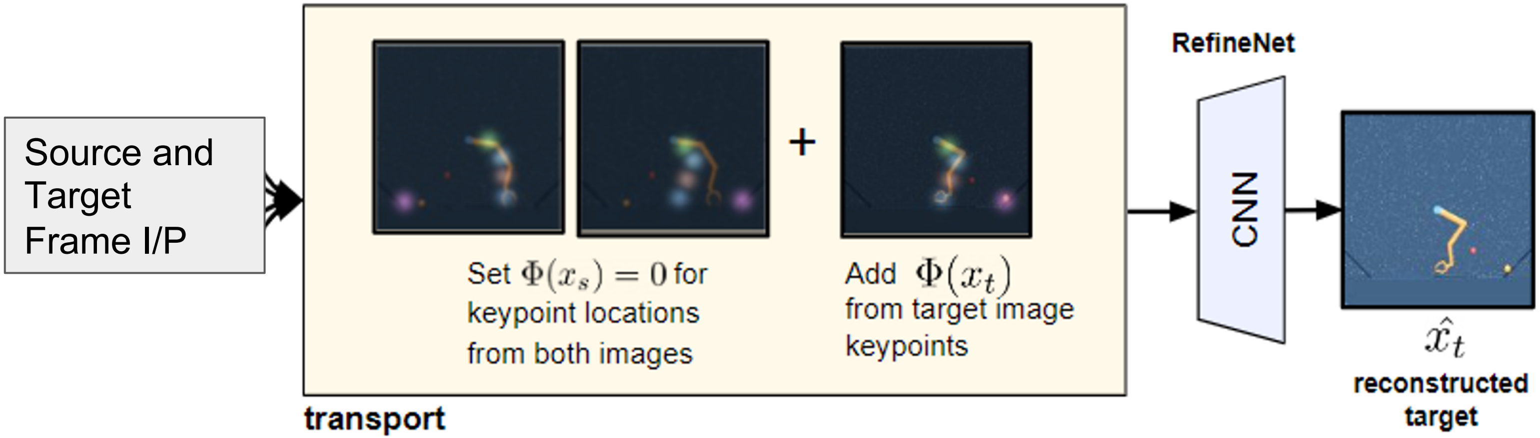
- Feature CNN to extract features
- Keynet to predict keypoints
- Features - $\Phi(\boldsymbol{x}_s), \Phi(\boldsymbol{x}_t) \in \mathbb{R}^{H' \times W' \times D}$
- Keypoints - $\Psi(\boldsymbol{x}_s), \Psi(\boldsymbol{x}_t) \in \mathbb{R}^{K \times 2}$



Transporter

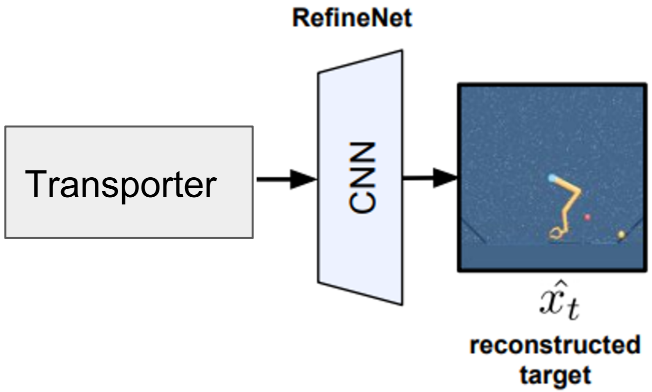
- The features in the source image at the target positions are replaced with the features from the target image.
- The features at the source position are set to zero.

$$\hat{\Phi}(\boldsymbol{x}_s,\boldsymbol{x}_t) \triangleq (1-\mathcal{H}_{\Psi(\boldsymbol{x}_s)})\cdot(1-\mathcal{H}_{\Psi(\boldsymbol{x}_t)})\cdot\Phi(x_s)+\mathcal{H}_{\Psi(\boldsymbol{x}_t)}\cdot\Phi(x_t)$$



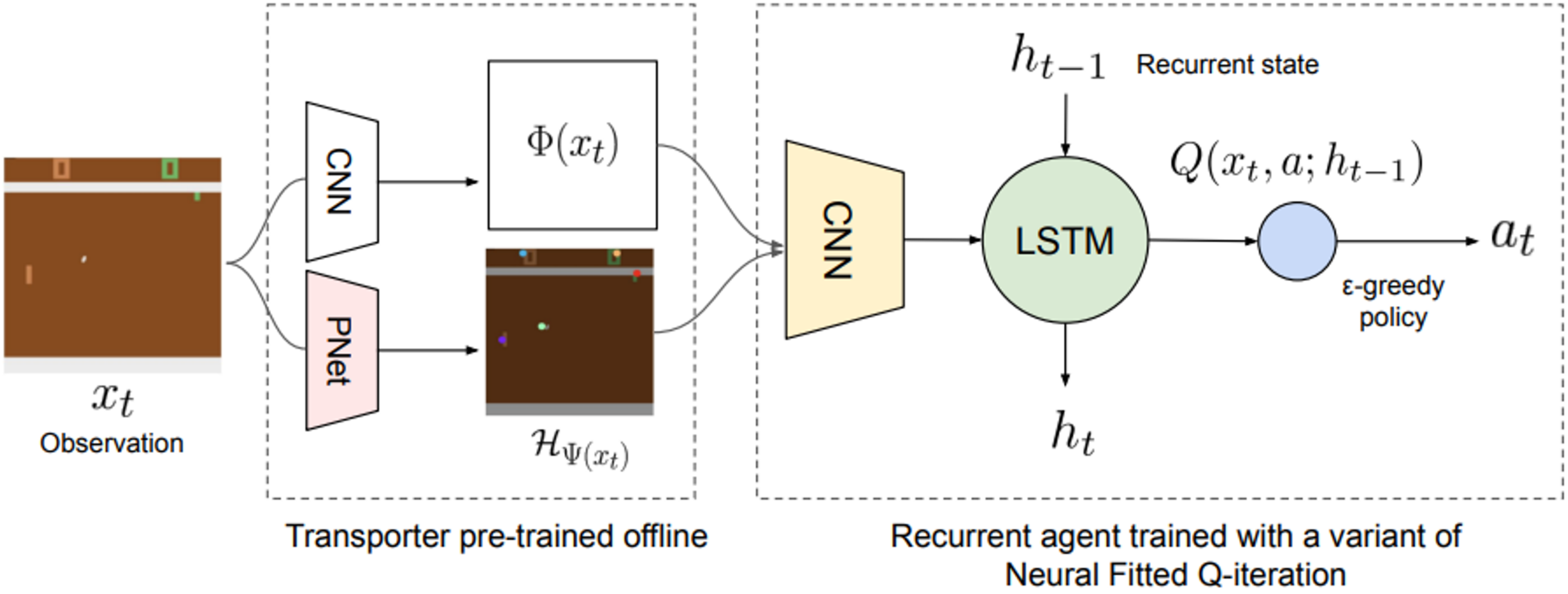
RefineNet

- Inpaint the missing features at the source position.
- Clean up the image around the target positions.
- Loss Function = $||\boldsymbol{x}_t - \hat{\boldsymbol{x}}_t||_2^2$ (pixel-wise squared-L2 reconstruction error)



Application: Data-efficient reinforcement learning

- Task-agnostic learning of keypoints enables faster learning of a policy.



Application: Keypoint-based options for efficient exploration

- Action space to explore is significantly made smaller with the use of keypoints.
- Transporter performs well with long temporal consistency.
- K × 4 intrinsic reward functions using the keypoint locations:

$$r_{i,1} = \Psi_x^i(\boldsymbol{x}_{t+1}) - \Psi_x^i(\boldsymbol{x}_t),\ r_{i,2} = \Psi_x^i(\boldsymbol{x}_t) - \Psi_x^i(\boldsymbol{x}_{t+1}),\ r_{i,3} = \Psi_y^i(\boldsymbol{x}_{t+1}) - \Psi_y^i(\boldsymbol{x}_t),\ r_{i,4} = \Psi_y^i(\boldsymbol{x}_t) - \Psi_y^i(\boldsymbol{x}_{t+1})$$

- Learn K × 4 Q functions {Qi,j |i ∈ {1,...,K},j ∈ {1,2,3,4}} to maximise each of the reward functions
- Most controllable keypoint:

$$\pi_{Q\text{gap}}(s) = \operatorname*{argmax}_i \frac{1}{4}\sum_{j=1}^{4} \max_a Q_{i,j}(s;a) - \min_a Q_{i,j}(s;a).$$

Experimental Setup

- Dataset - Atari ALE and Manipulator

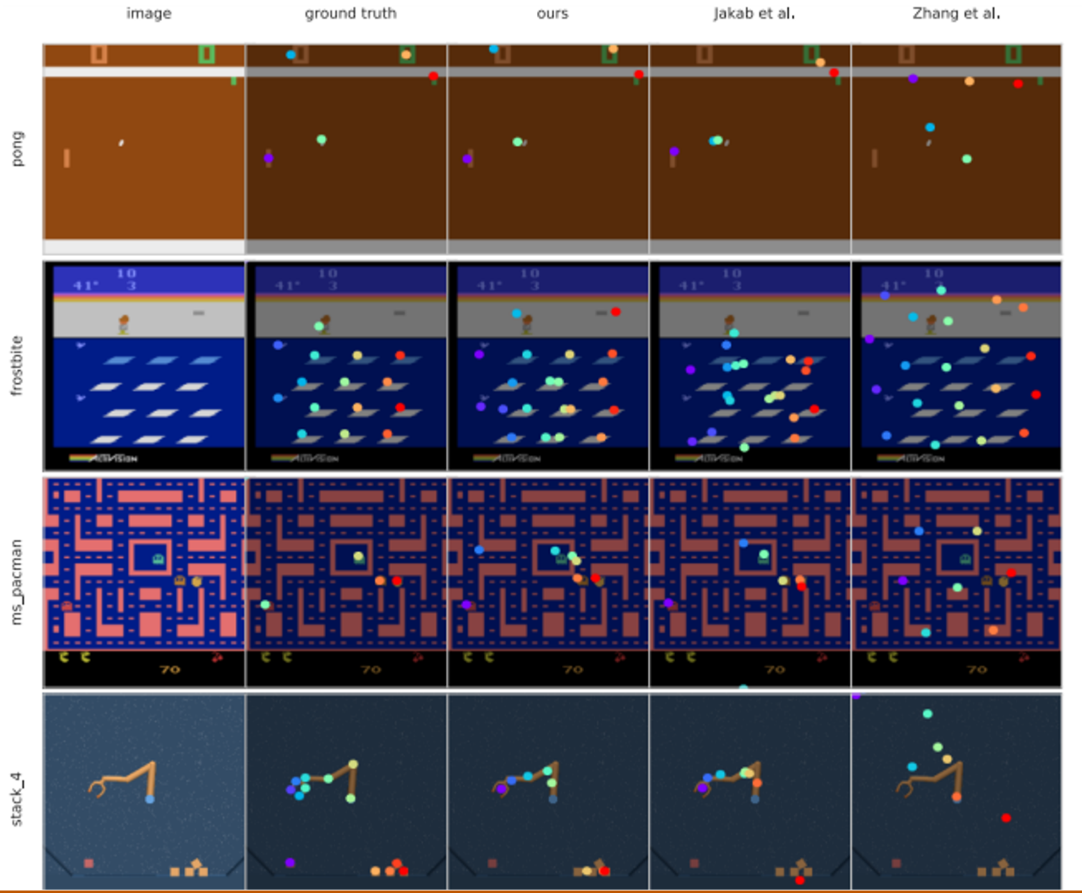
(1) For evaluating long-term tracking of object keypoints section — pong, frostbite, ms_pacman, and stack_4 (manipulator with blocks)

(2) For data-efficient reinforcement learning — random exploration on the Atari game.

(3) For keypoints based efficient-exploration; one of the most difficult exploration game — montezuma revenge, along with ms_pacman and seaquest.

- The source and target frames are sampled randomly within a temporal offset of 1 to 20 frames.

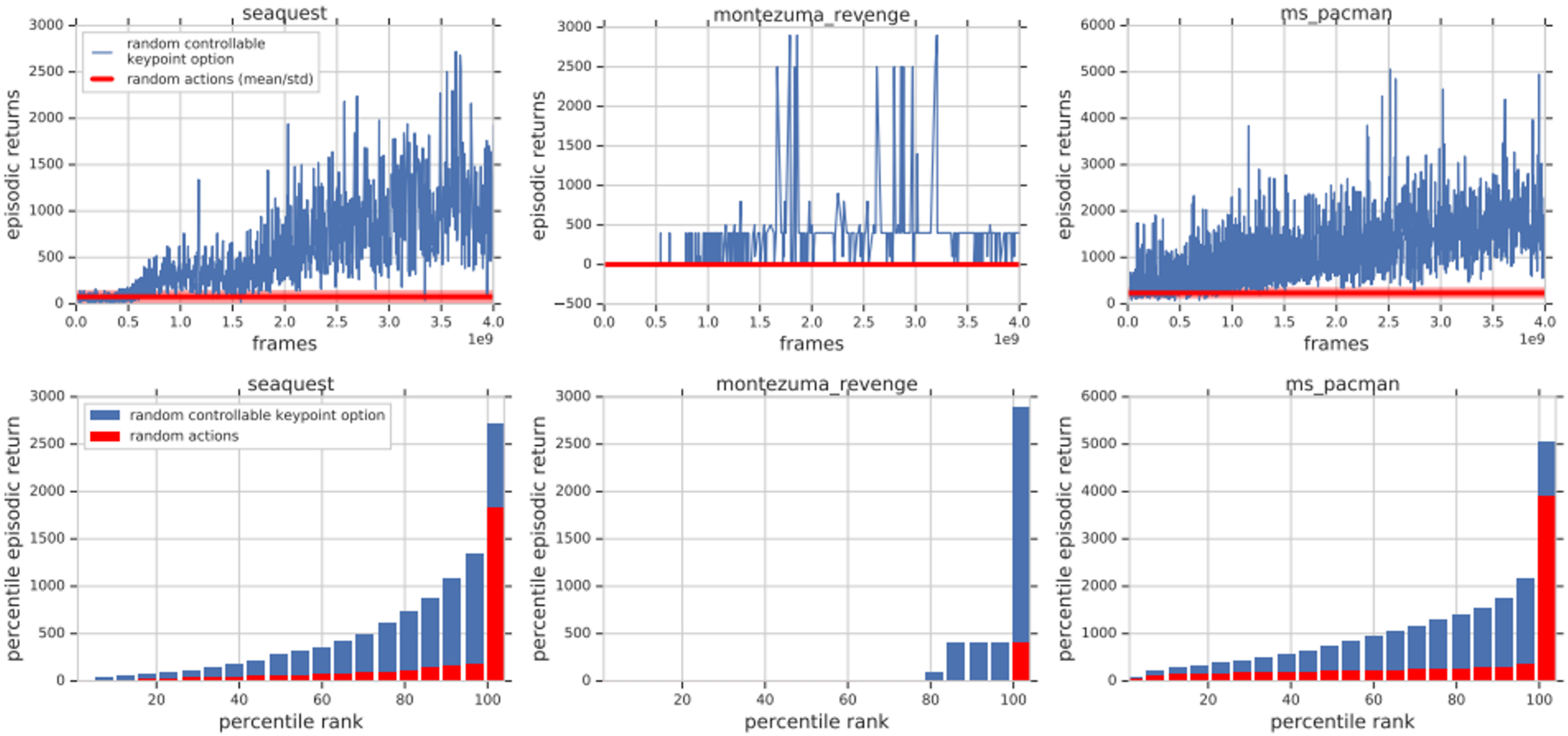
Results: Evaluating Object Keypoint Predictions



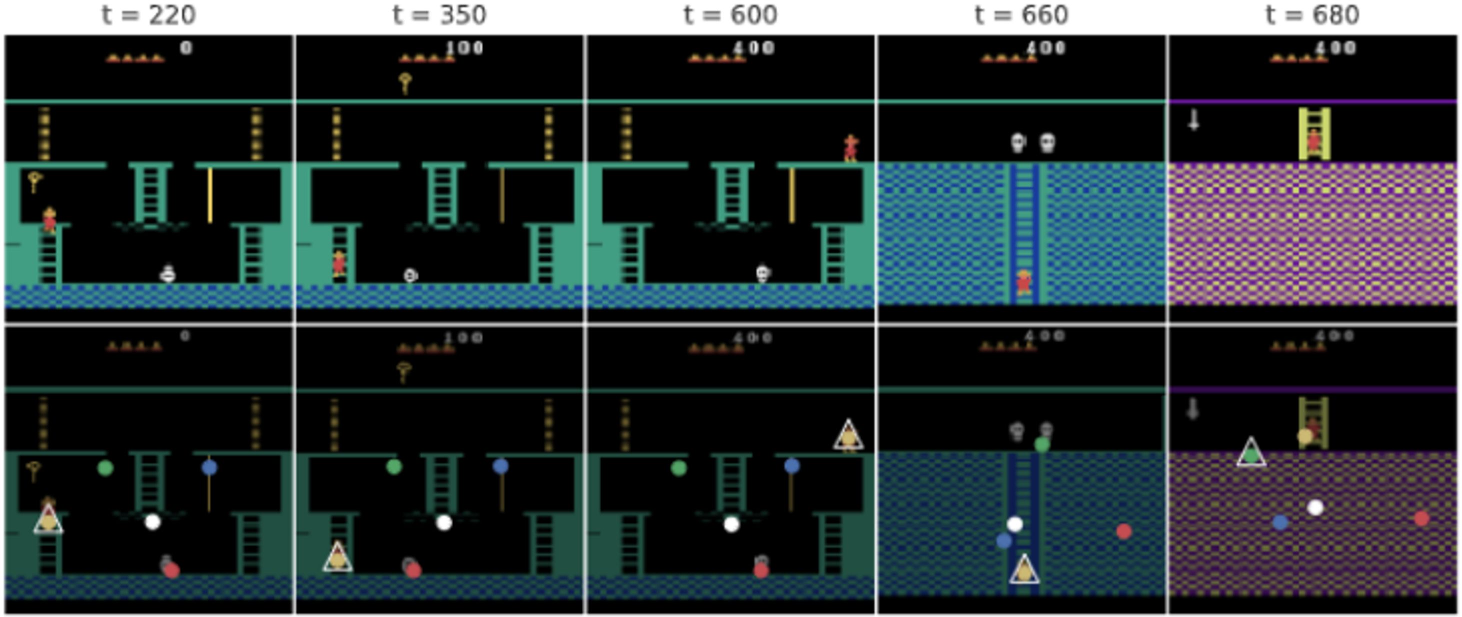
Results: Data-efficient Reinforcement Learning on Atari

Game	KeyQN (ours)	SimPLe	Rainbow	PPO (100k)	Human	Random
breakout	19.3 (4.5)	12.7 (3.8)	3.3 (0.1)	5.9 (3.3)	31.8	1.7
frostbite	388.3 (142.1)	254.7 (4.9)	140.1 (2.7)	174.0 (40.7)	4334.7	65.2
ms_pacman	999.4 (145.4)	762.8 (331.5)	364.3 (20.4)	496.0 (379.8)	15693.0	307.3
pong	10.8 (5.7)	5.2 (9.7)	-19.5 (0.2)	-20.5 (0.6)	9.3	-20.7
seaquest	236.7 (22.2)	370.9 (128.2)	206.3 (17.1)	370.0 (103.3)	20182.0	-20.7

Results: Efficient Exploration with Keypoints



Results: Efficient Exploration with Keypoints



Overview of Results

- Transporter outperforms state-of-the-art keypoint generation model.
- Able to learn stable-keypoints without task-specific reward functions.
- Accurately track objects over long-temporal sequences.
- By efficiently reducing the action space by learning using keypoints, the authors demonstrated drastic reduction in search complexity and thus, efficient exploration.

Limitations

- Cannot handle moving backgrounds.
- No experimentation or analysis for real-world scenarios, which is essential for robotics.
- In games like ms_packman and frostbite the model did showcase a decrease in tracking keypoints over time. The model is not immune against temporal issues, and there is potential for further improvement.
- Lack of detail regarding real-time inference speed for detecting and tracking keypoints, which again, is a critical component of robotics.

Future work

- Improvement of performance with moving background.
- Transfer of method to real-world application.
- Research to integrate attention-based architecture like Transformer, to extrapolate more relevant keypoints over long temporal periods.

Extended readings

[1] Burgess, C.P., Matthey, L., Watters, N., Kabra, R., Higgins, I., Botvinick, M., & Lerchner, A. (2019). MONet: Unsupervised Scene Decomposition and Representation. ArXiv, abs/1901.11390. - 3D reconstruction of images using recurrent attention network

[2] Jakab, T., Gupta, A., Bilen, H., & Vedaldi, A. (2020). Self-Supervised Learning of Interpretable Keypoints From Unlabelled Videos. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 8784-8794. - Learn the pose of an object from a single image using unlabelled videos.

[3] Kipf, T., Pol, E.V., & Welling, M. (2020). Contrastive Learning of Structured World Models. ArXiv, abs/1911.12247. - Contrastive approach for representation learning in environments with compositional structure.

Summary

- Problem Setting - Accurate object representation for control and RL
- Prior Limitations - Usually task-specific, focussed on classification and segmentation. For RL and control domains, precise geo-spatial representation of an object is required.
- Key Insights of Proposed Method -

1. Accurate over long-temporal sequences when compared to contemporary methods.
2. Task-agnostic. Take input of keypoints for formulation of policies.
3. Using keypoints provides an efficient action-space for exploration.

- Result - SOTA performance, much better than baseline in almost all comparison metrics.

THANK YOU