PermaKey

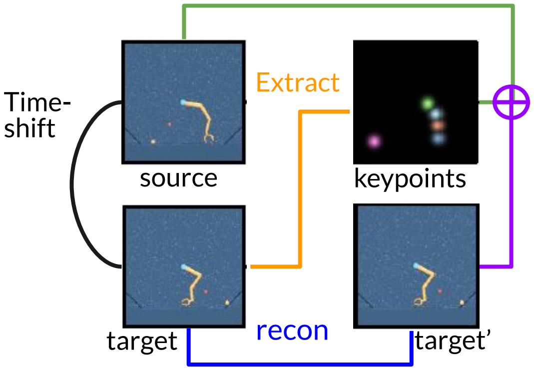
Unsupervised Object Keypoint Learning using Local Spatial Predictability

<u>Anand Gopalakrishnan</u>, Sjoerd van Steenkiste, and Jürgen Schmidhuber

Introduction

- **Goal**: learn structured representations (object keypoints) of object-parts
  - Compact format that distills essential information
  - Easier for learning and generalization

- Previous keypoint approaches
  - Based on information bottleneck
  - Overly focused on extrinsic properties



“Transporter” by Kulkarni et. al [1]

[1] Kulkarni et al., “Unsupervised Learning of Object Keypoints for Perception and Control.” *NeurIPS* 2019.

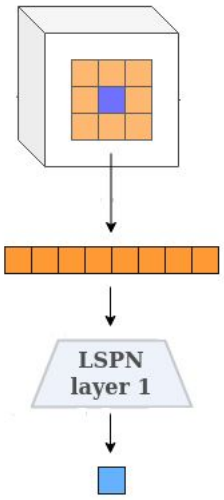
PermaKey: Local Spatial Prediction

- **Objects are …**
    - *Modular* building blocks -> self-contained and reusable
    - Local regions with *high internal predictive structure*

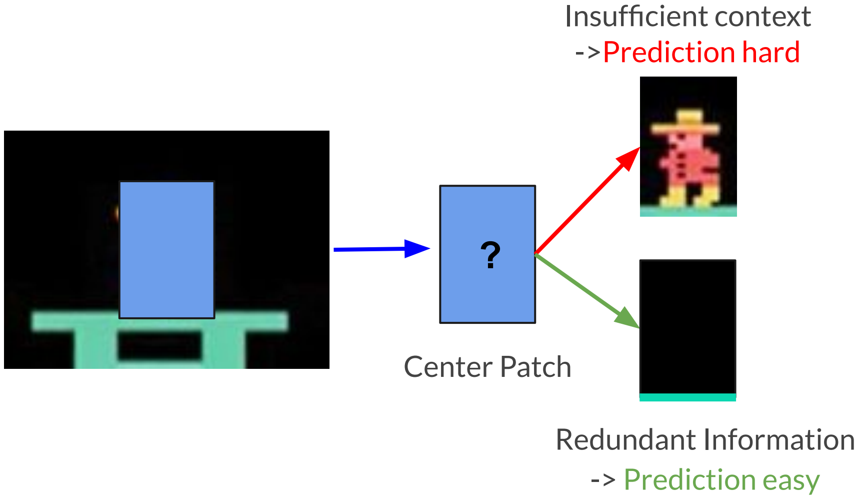
- Focus on *“Local predictability”* -> intrinsic object property

- Local spatial prediction (LSP) problem
    - **Task ->** predict center patch given neighbours

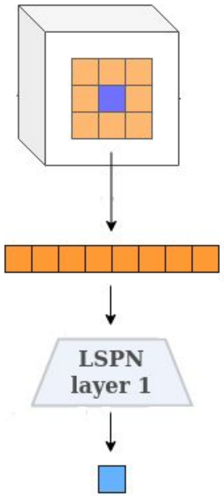
Module 1: LSP



PermaKey: Local Spatial Prediction

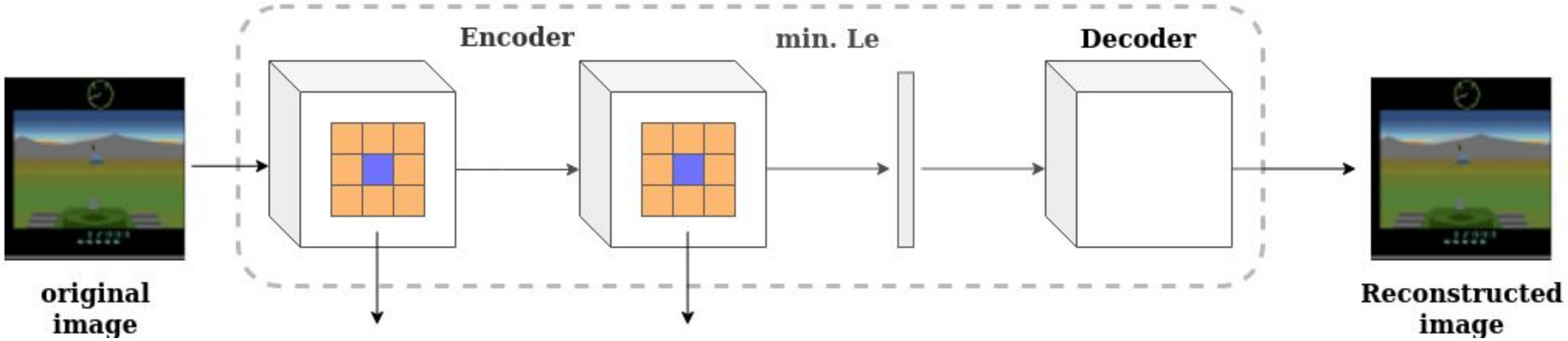


Module 1: LSP



- Neighbourhood with same object -> Prediction easy
- Errors in prediction -> salient objects

PermaKey: Spatial Feature Embedding

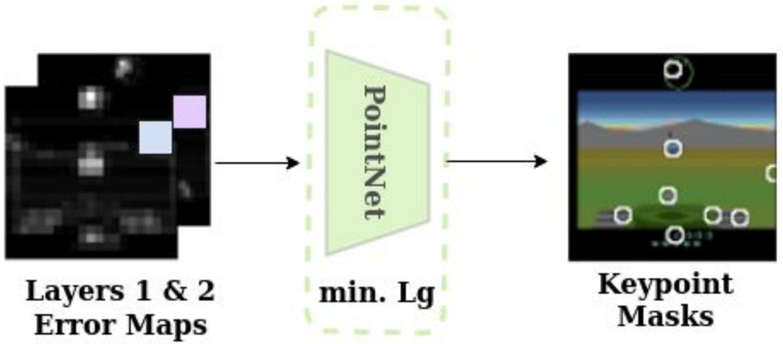


Module 2: VAE

- Learn features for prediction task using a VAE trained with ELBO.
- Layer choice -> tradeoff b/w expressivity vs locality.
  - Lower layers -> high spatial resolution but poor feature descriptors
  - Deeper layers -> low spatial resolution but rich feature descriptors

PermaKey: Error Maps -> Keypoints

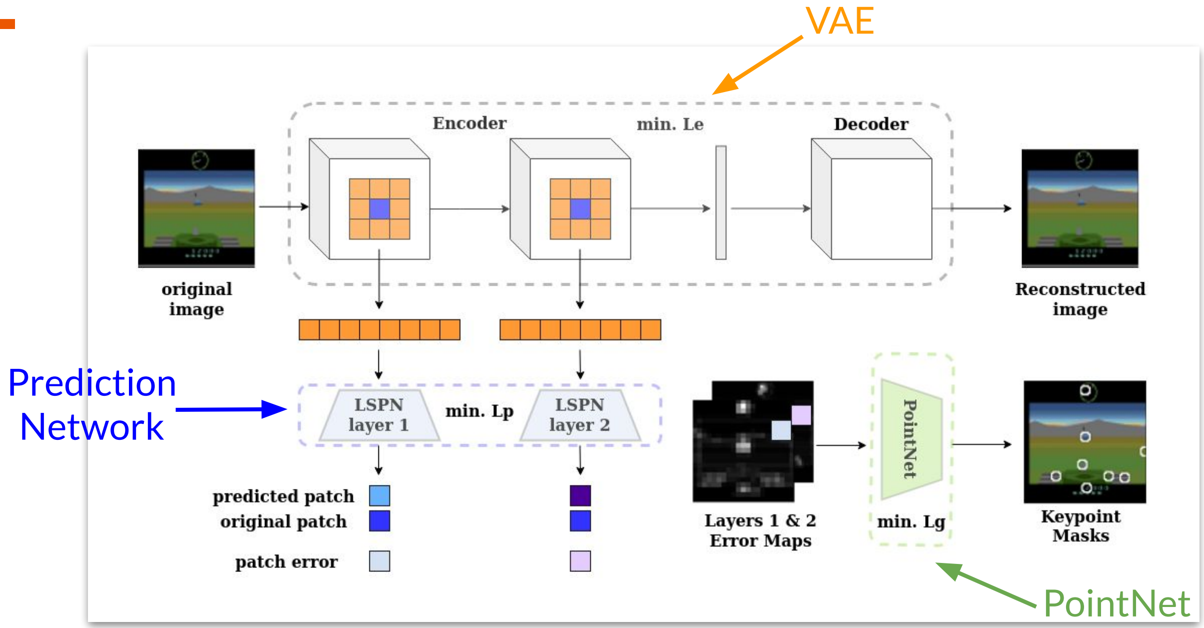
- Extract largest error clusters corresponding to objects parts as keypoints
- Use PointNet *(Jakab et. al [2])* to reconstruct error maps using MSE objective



Module 3: PointNet

[2] Jakab et. al, "Unsupervised Learning of Object Landmarks through Conditional Image Generation". NeurIPS 2018.

PermaKey: Overview

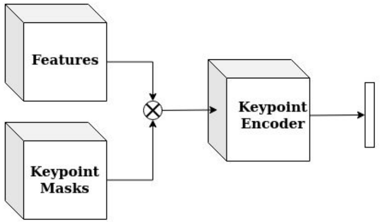


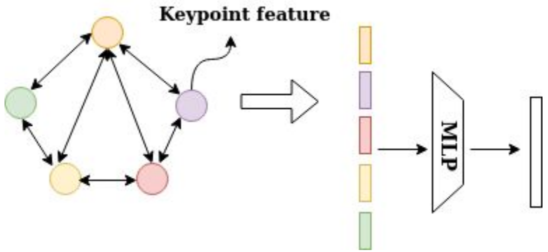
PermaKey system with all 3 modules

Keypoints-based Atari Agent

- How to incorporate keypoints into a state representation?
  - Represent relational effects b/w agent & entities
  - CNN-based keypoint encoding -> limited (spatially) relational encoding
  - GNN-based keypoint encoding -> more flexible relational encoding

CNN-based





GNN-based

- RL algorithm
  - Recurrent deep Q-learning w/ double-Q [4], target network and 3-step returns
  - Train on low-data regime using protocol as in [3]

[3] Kaiser, Lukasz, et al. "Model-based reinforcement learning for atari.", *ICLR 2020.*
[4] Hausknecht, Matthew, et al. "Deep recurrent q-learning for partially observable mdps.", *AAAI 2015.*

Results

PermaKey qualitatively similar to Transporter

PermaKey keypoints lie on salient objects i.e. ice floats, player

Frostbite

Battlezone

Transporter captures uninformative regions as keypoints

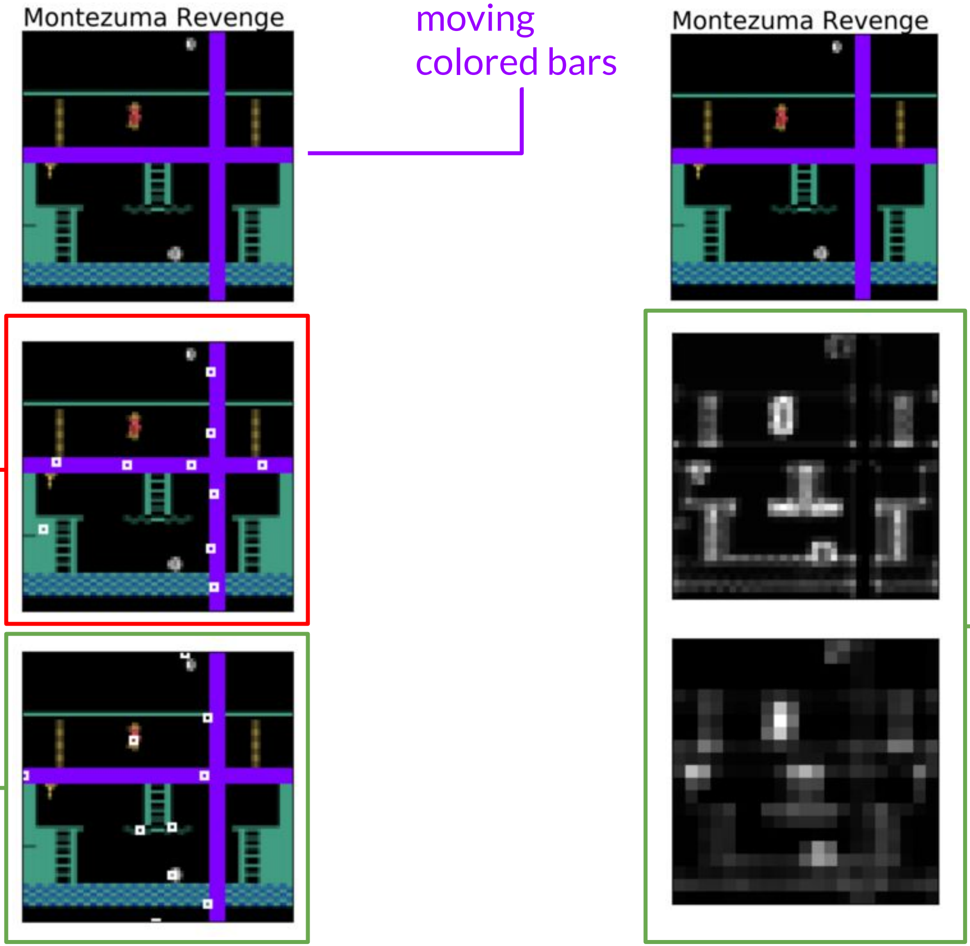
PermaKey captures enemy and player tank parts as keypoints

Atari

Results

Transporter places keypoints on distractor bars

PermaKey places keypoints on salient objects



PermaKey error maps show no spikes at distractor locations

Noisy Atari

Atari RL Quantitative Results

Game	Rainbow	PPO	SimPLE	Transporter (orig.)	Transporter (re-imp.)	Transporter -GNN	PKey-CNN (ours)	PKey-GNN (ours)
Battlezone	3363.5 (523.8)	5300.0 (3655.1)	5184.4 (1347.5)	N/A	N/A	N/A	10266.7 (3272.1)	**12566.7** (3297.9)
MsPacman	364.3 (20.4)	496.0 (379.8)	762.8 (331.5)	999.4 (145.4)	983.0 (806.3)	838.3 (537.3)	**1038.5** (417.1)	906.3 (382.2)
Seaquest	206.3 (17.1)	370.0 (103.3)	370.9 (128.2)	236.7 (22.2)	455.3 (160.6)	296.0 (82.5)	**520.0** (159.9)	375.3 (75.4)
Frostbite	140.1 (2.7)	174.0 (40.7)	254.7 (4.9)	388.3 (142.1)	263.7 (14.9)	465.0 (442.1)	310.7 (150.1)	**657.3** (556.8)

- PermaKey-based agents > Transporter analogues
- PermaKey-based agents > Rainbow & PPO (model-free) and SimPLE (model-based)
- GNN-based keypoint encoding sometimes better than CNN-based ones.

More RL Results....

- PermaKey-based agents more robust than Transporter on "Noisy" Atari environments.

No Noise

Game	Transp. (re-imp.)	PKey + CNN
Seaquest	455.3 (160.6)	**520.0** (159.9)
MsPacman	983.0 (806.3)	**1038.5** (417.1)

With Noise

Game	Transp. (re-imp.)	PKey + CNN
Seaquest	357.3 (99.10)	**562.0** (114.78)
MsPacman	470.0 (212.90)	**574.7** (290.07)

- Number of keypoints ablation: PermaKey-based agent > Transporter across varied keypoint range.

MsPacman	5 keypoints	7 keypoints	10 keypoints
Transp. (re-imp.)	923.0 (433.95)	983.0 (806.3)	907.0 (317.41)
PKey + CNN	**1004.3** (319.15)	**1038.5** (417.1)	**1003.3** (313.07)

Conclusions

- *PermaKey*
    - Leverages "*local predictability*" -> find salient regions and extract keypoints
    - Intrinsic property of object -> robust to distractors
    - PermaKey-based RL agents -> excel on sample-efficiency benchmark

- **Future Directions:**
    - Quantitative evaluation metric -> objectively evaluate different models
    - Scaling up current unsupervised keypoint -> more complex visual scenes

Please visit us at poster session 5 to find out more!!

Paper *https://arxiv.org/abs/2011.12930*

Code *https://github.com/agopal42/permakey*

@agopal42