Enhancing Deep Reinforcement Learning with Compressed Sensing-based State Estimation

Shaswot Shresthamali, Masaaki Kondo

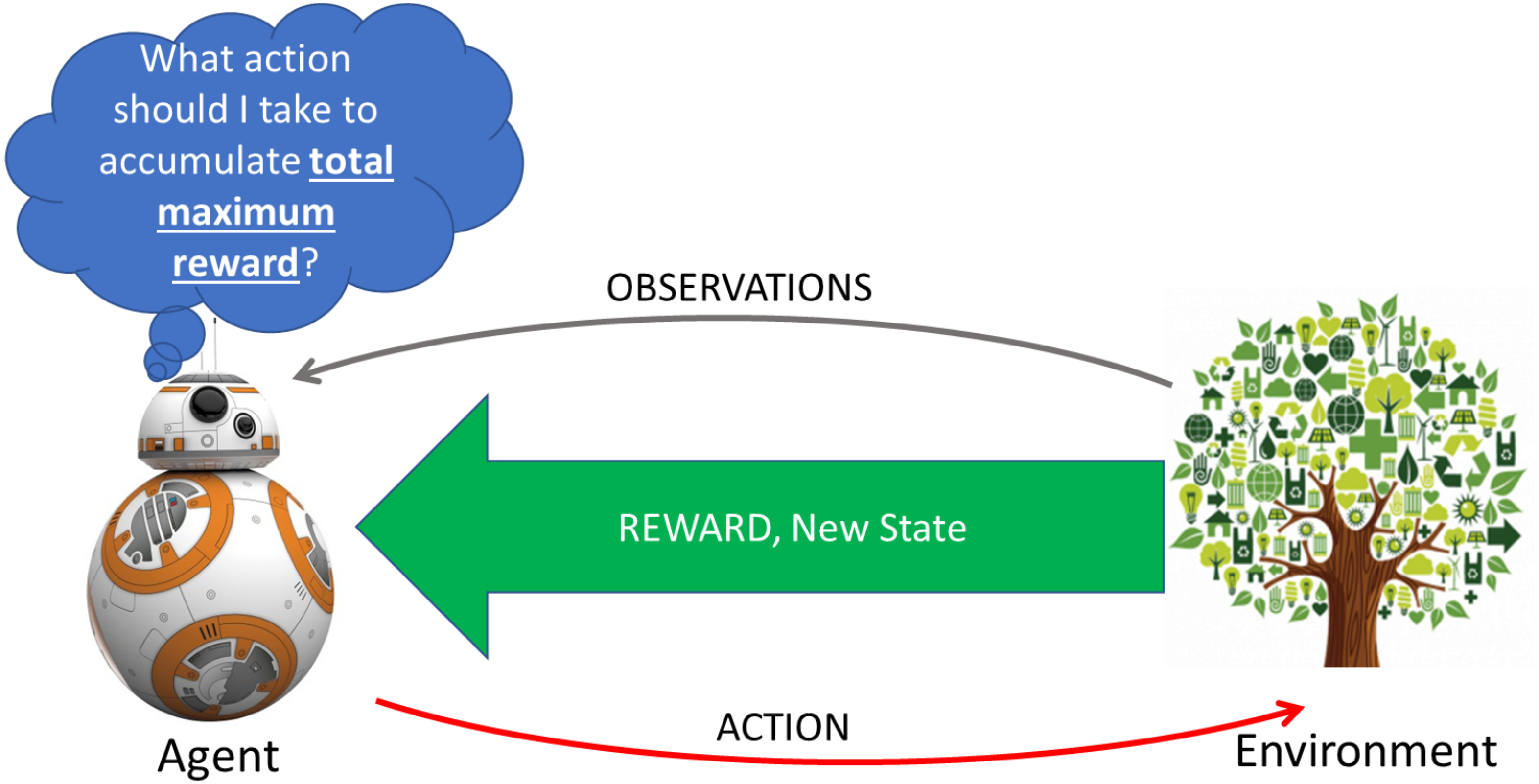
Kondo Laboratory
Department of Information and Computer Science
Keio University

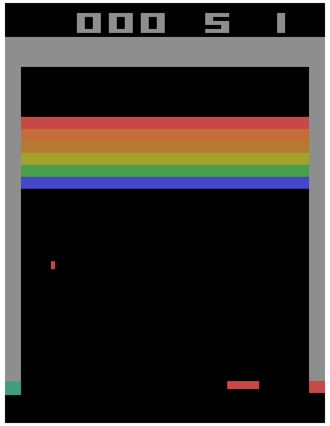
MCSoC 2023, Singapore

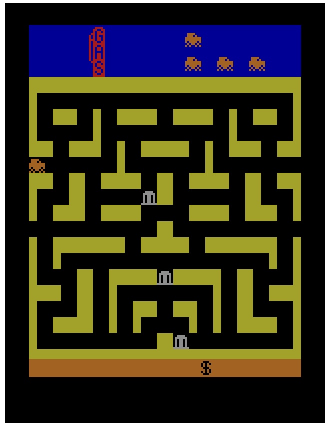
19-Dec-2023

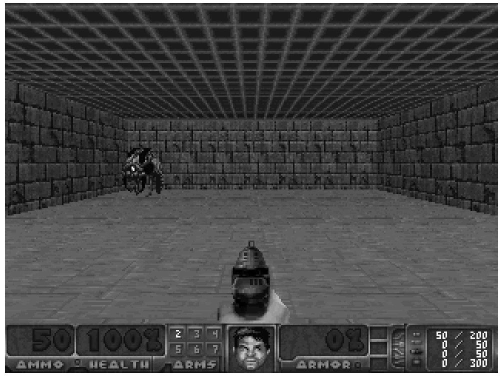
Reinforcement Learning (RL)

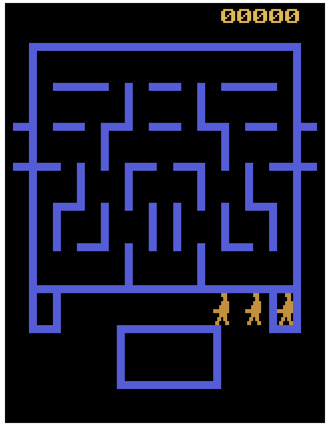
- Sequential Decision-making
- Feedback is provided using scalar rewards

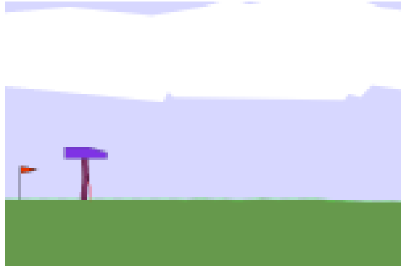










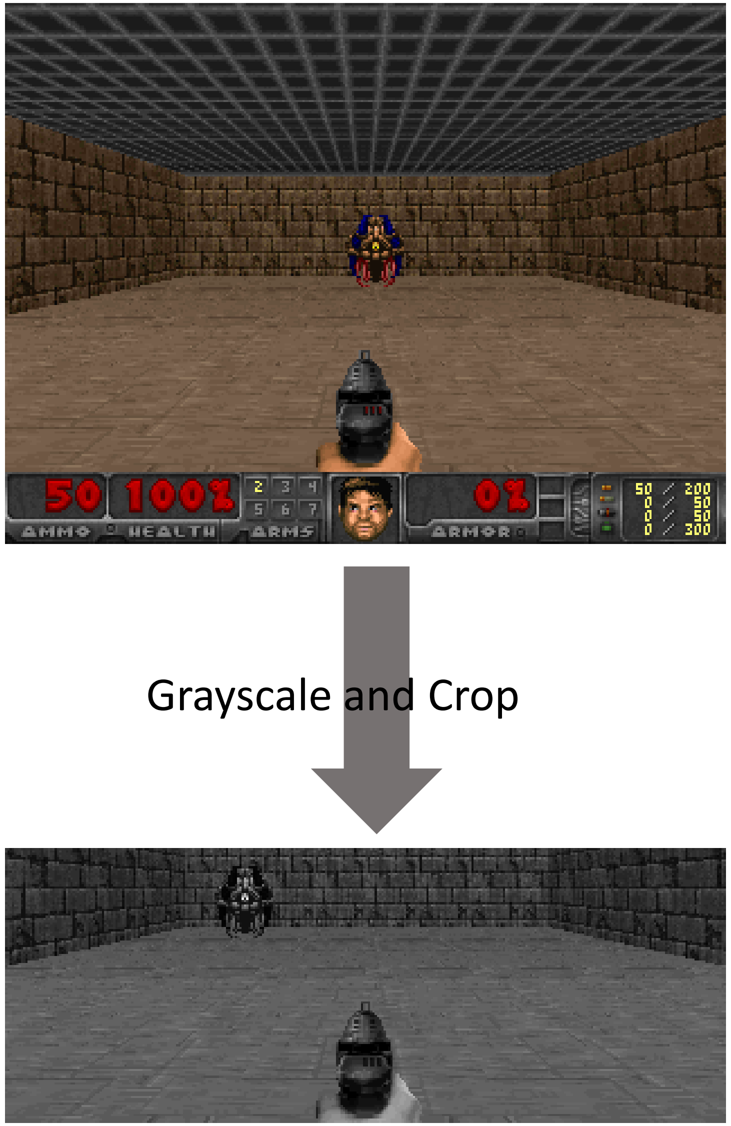


Compressed Sensing + RL

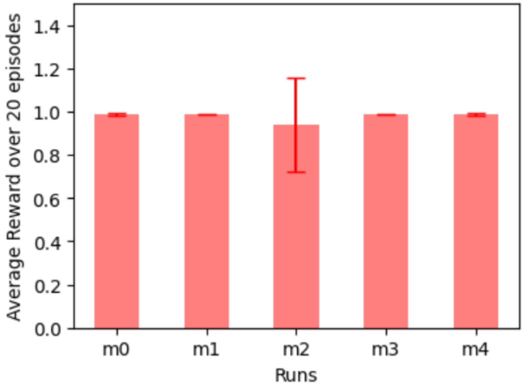
- Adaptive control using RL has become very popular in the past years
  - IoT, AlphaGo/Star/Fold, Plasma Control, Scheduling jobs/power in embedded systems
- RL suffer from missing state information due to
  - Sensor failures
  - Data transmission errors
  - Other noise sources
- Can we recover performance of RL agents even in the presence of missing state data?
- We use Compressed Sensing (CS) to recover/reconstruct missing sensor input for robust RL.

The ViZDoom Environment

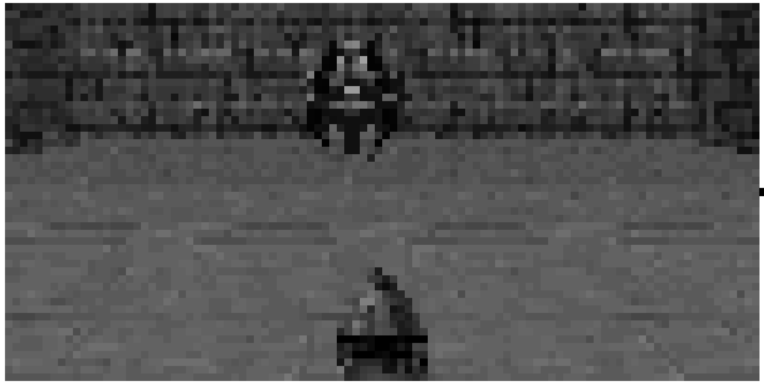
- First person shooter game - BASIC scenario
- Objective: shoot the monster
- Actions
  - go-left,
  - go-right,
  - shoot
- Rewards
  - -1 reward at each timestep
  - Successful hit: +101
  - Miss: -5
- State space
  - Pixel information: grayscaled and cropped



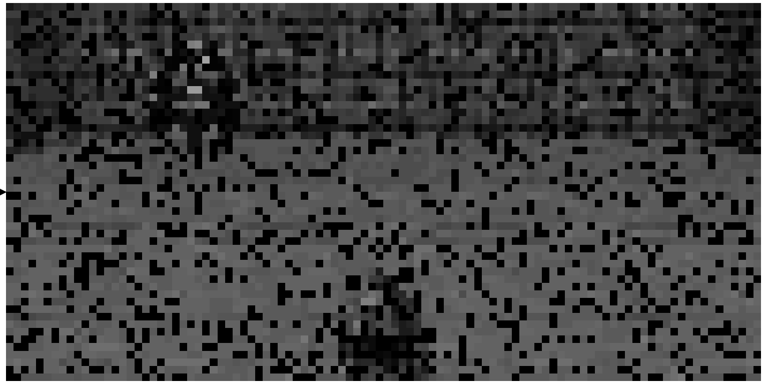
Degradation due to missing pixels

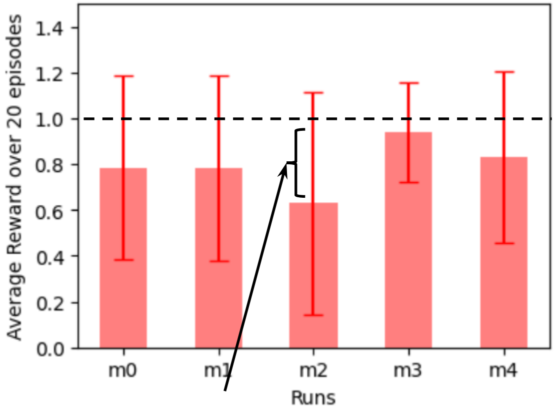


Original Pixel Observation



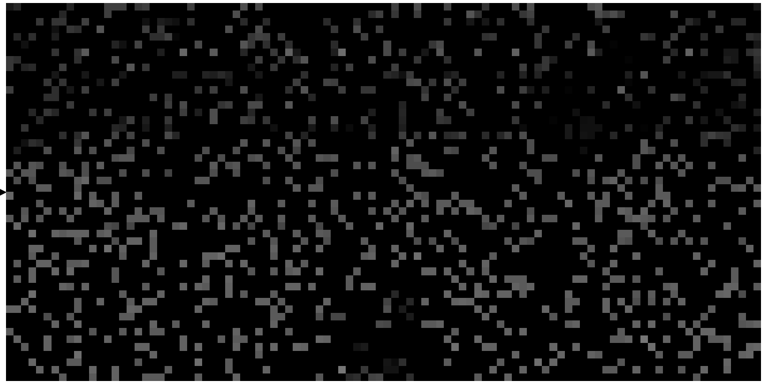
Noisy Observation, noise = 0.2

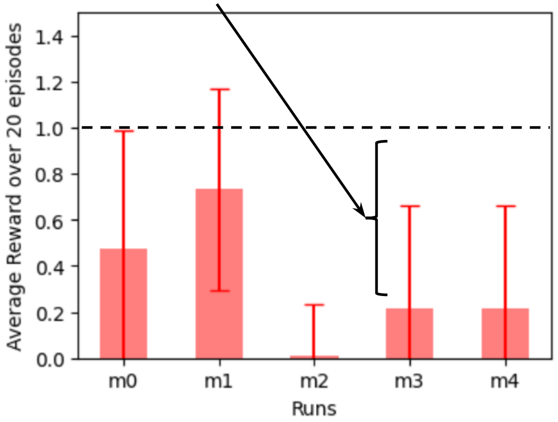




Degradation due to noise

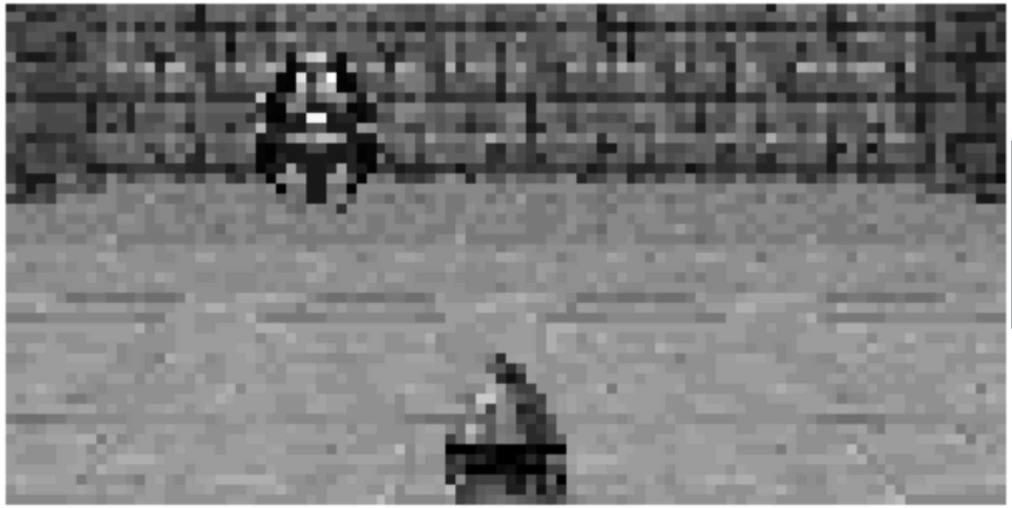
Noisy Observation, noise = 0.8



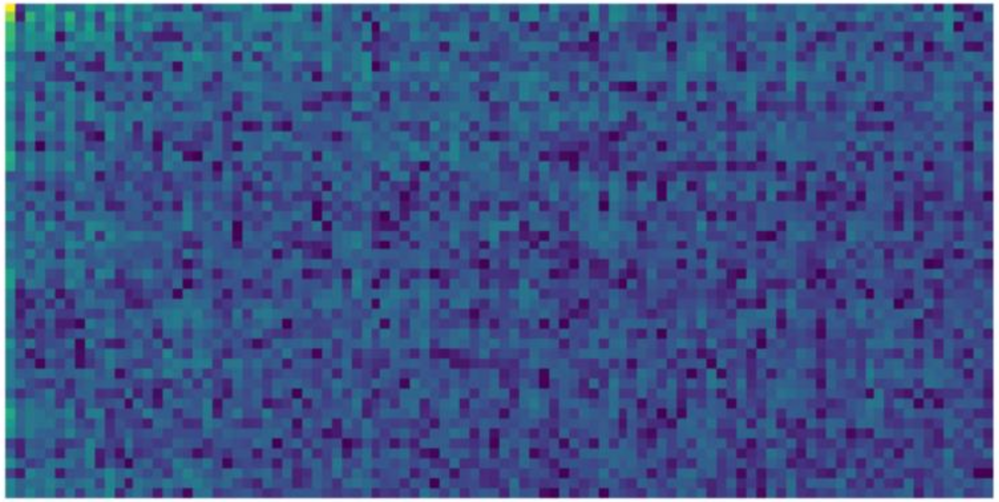


noise = probability of pixel being blacked out (i.e., pixel data is missing)

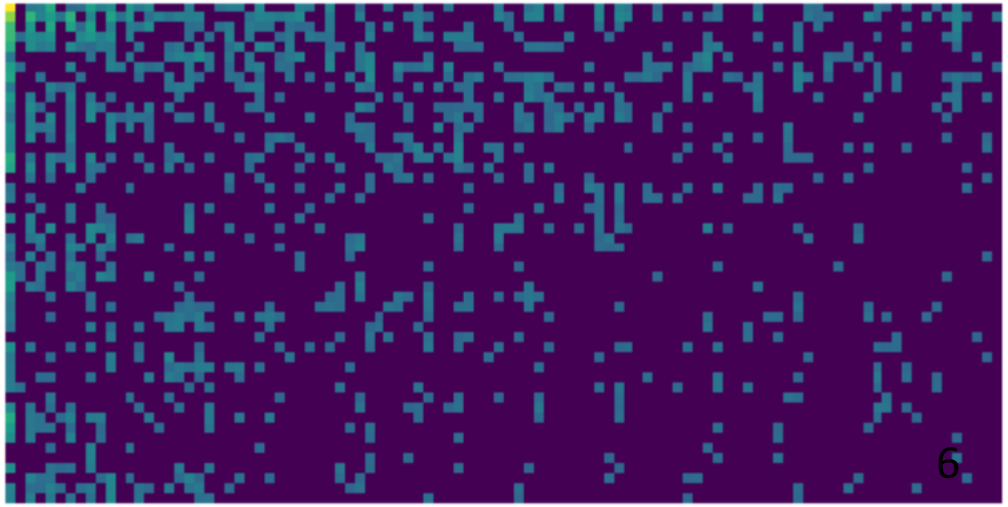
Image Compression: Review



DCT (Discrete Cosine Transform)

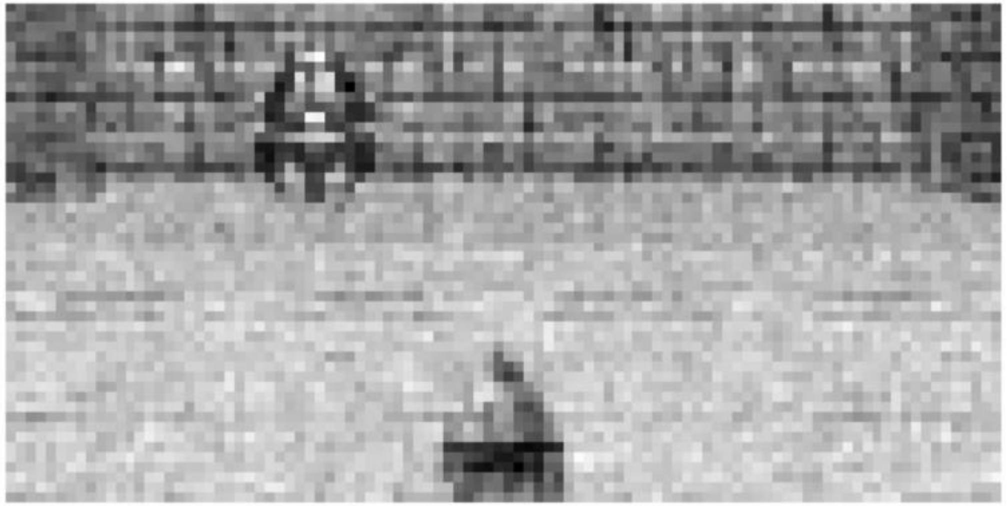


Keep only 20% of the largest coefficients (Compress by 80%)

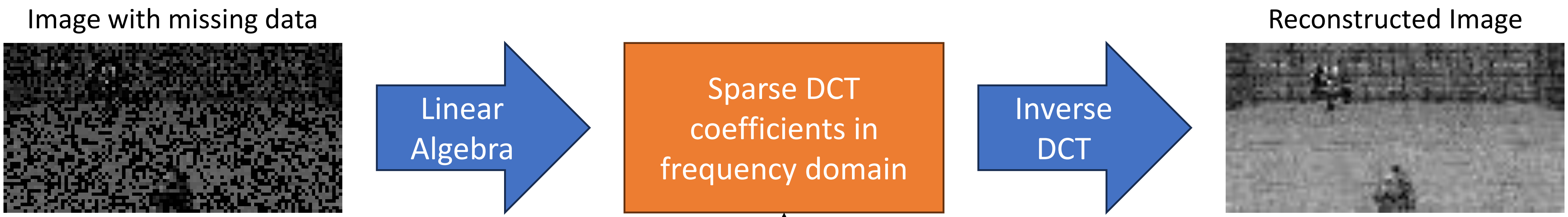


Inverse DCT

The original image can be reconstructed almost perfectly

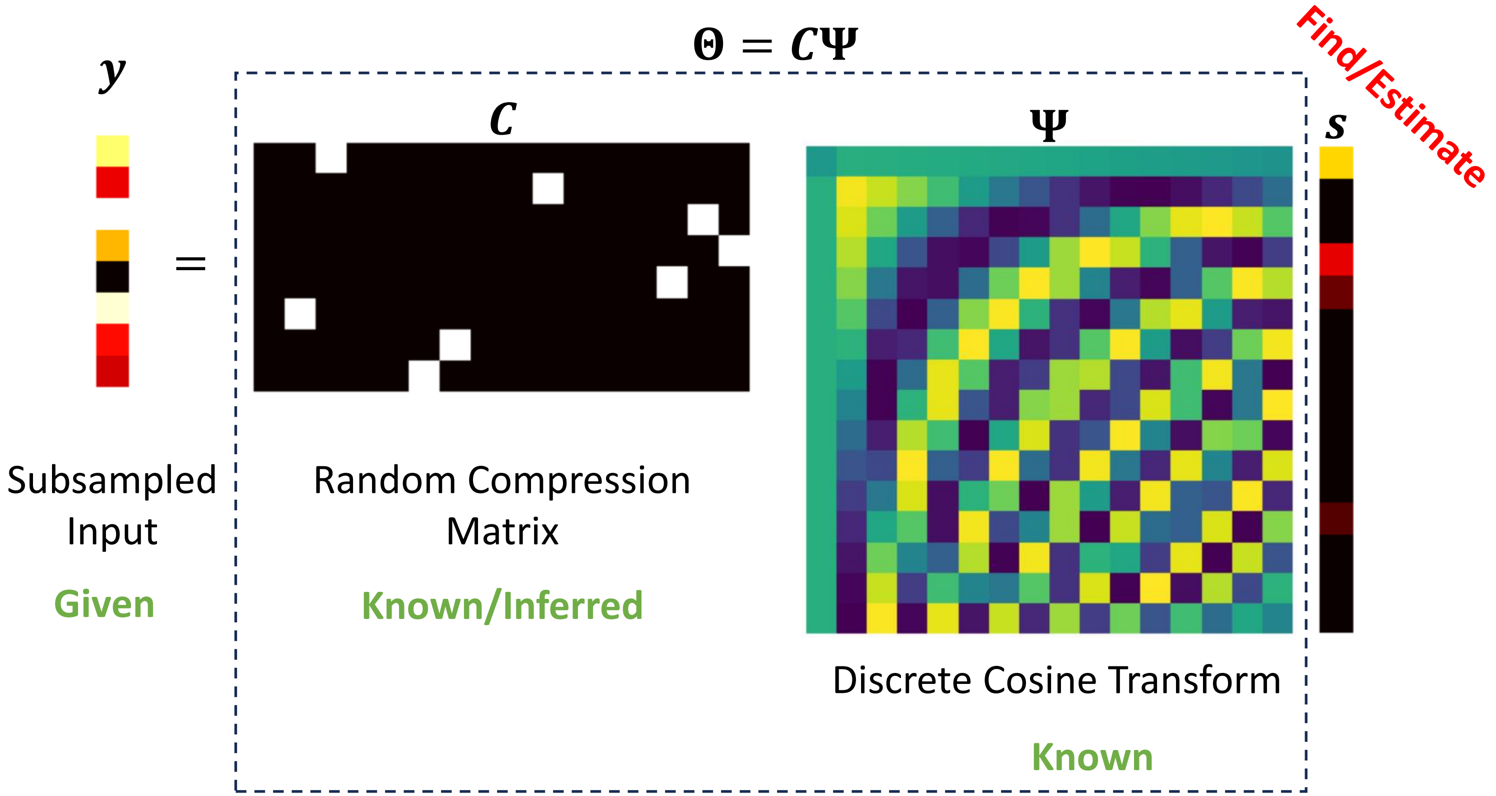


Magic Behind Compressed Sensing



The trick is **to determine the sparse DCT coefficients** from the undersampled image.

Compressed Sensing Problem Statement



Once we can obtain an estimate of $\mathbf{s}$ as $\hat{\mathbf{s}}$, we can reconstruct an estimate of the original image by taking the inverse DCT.

Underdetermined system of linear equations

- Infinitely large number of solutions
- The trick
  - Find the ***sparsest*** solution
  - Use L1 regularization function
    - Better behavior than L0

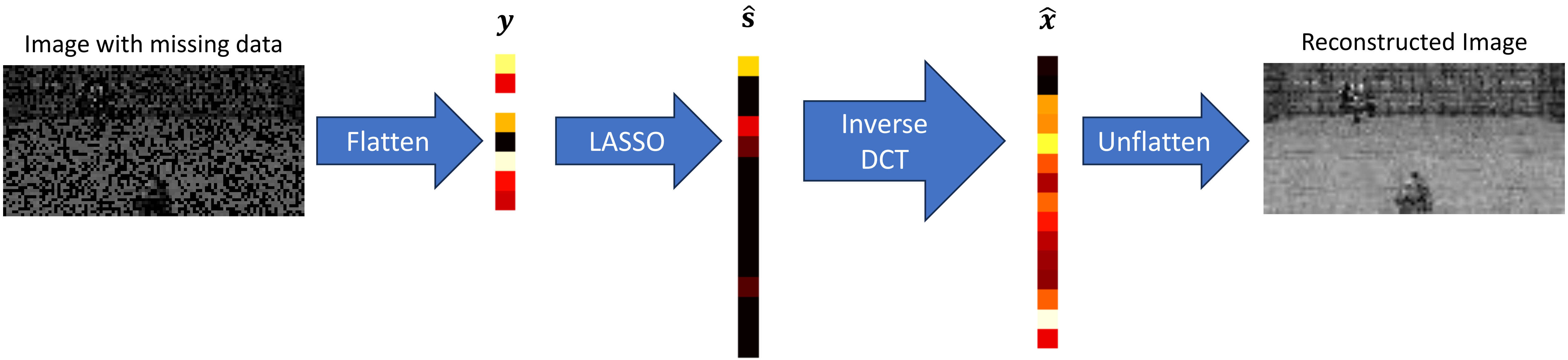
$$\hat{\mathbf{s}} = \underset{\mathbf{s}}{\operatorname{argmin}} \, ||\mathbf{\Theta s} - \mathbf{y}||_2 + \alpha||\hat{\mathbf{s}}||_1$$

where,

$$||\mathbf{s}||_1 = \sum_n^{k=1} |s_k|$$

Use **LASSO (Least Absolute Shrinkage and Selection Operator)** to solve the optimization problem and find $\hat{\mathbf{s}}$

Takeaway and Caveats



$\boldsymbol{\Theta} = \boldsymbol{C\Psi}$ must satisfy the **Restricted Isometry Property (RIP)**

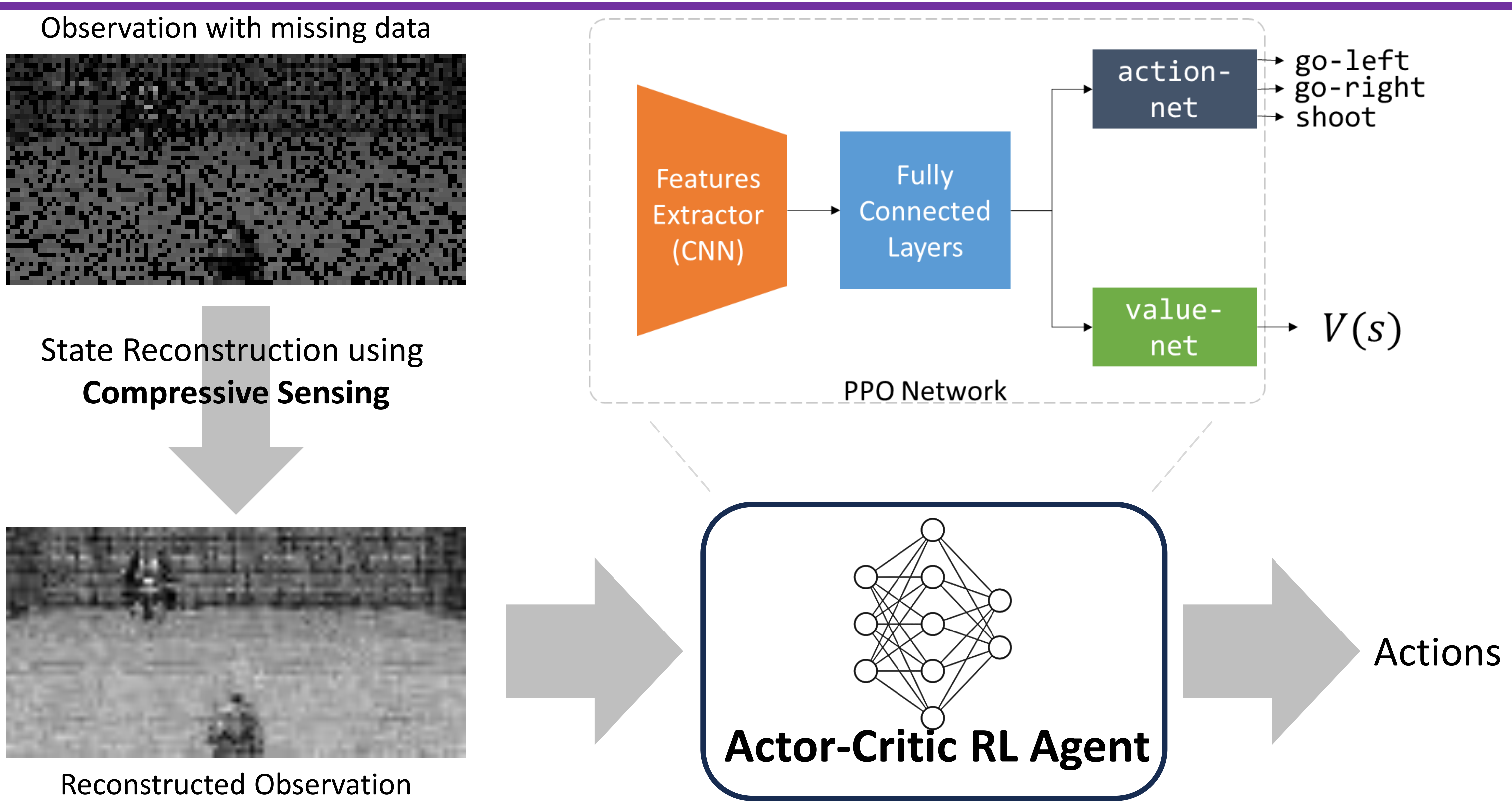
- <u>$\boldsymbol{C}$ and $\boldsymbol{\Psi}$ should be non-coherent</u>
  - Usually satisfied if $\boldsymbol{C}$ is a random sampling matrix
- Distances between two signals are preserved after being projected by $\boldsymbol{\Theta}$

If $\boldsymbol{s}$ is K-sparse (i.e., it has K non-zero values), then the **<u>number of measurements $\boldsymbol{p}$, should be sufficient large</u>** on the order of

$$p \approx \mathcal{O}(K \log(n/K)) \approx k_1 K \log(n/K) \qquad (3)$$

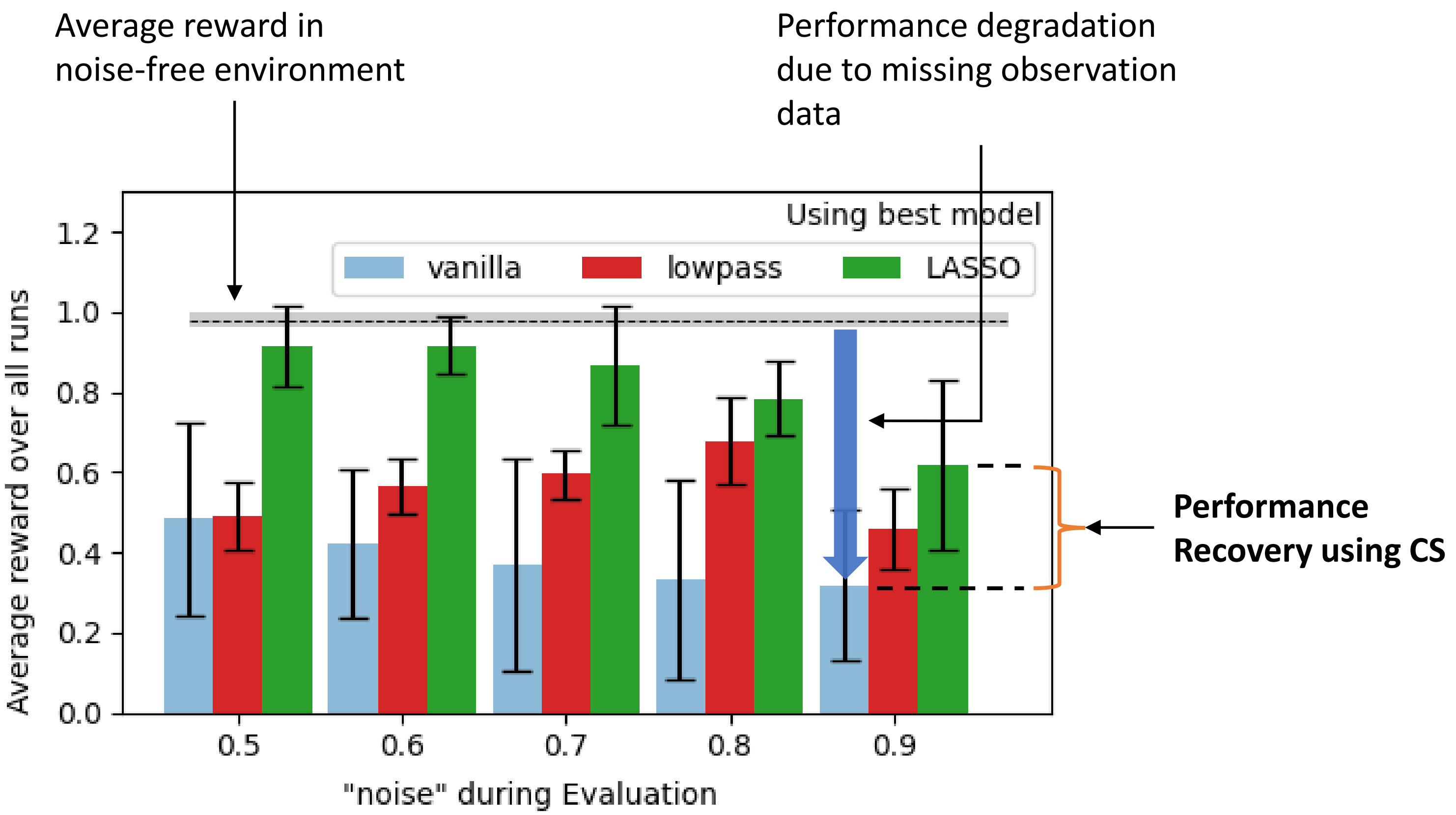
where the constant multiplier k_1 depends on how incoherent $\mathbf{C}$ and $\boldsymbol{\Psi}$ are.

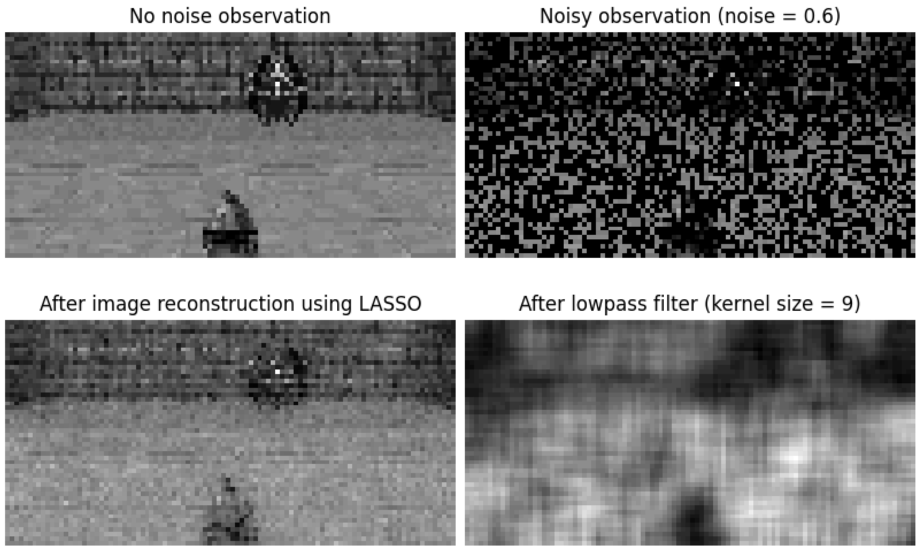
Proposed Framework: RL + CS



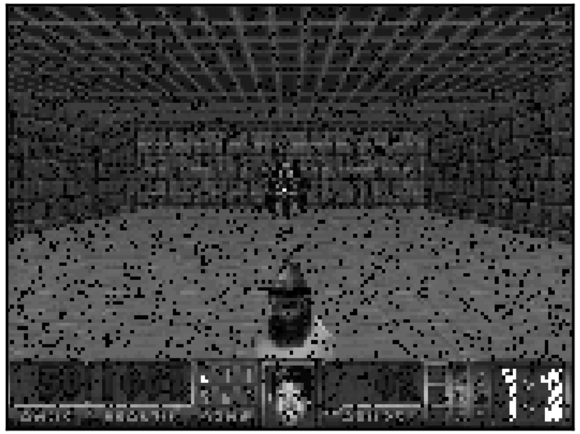
Results: Performance Recovery with CS

- Agent trained in noise-free environment
- Evaluated in noisy environment
  - Input recovery using `lowpass` filter and LASSO



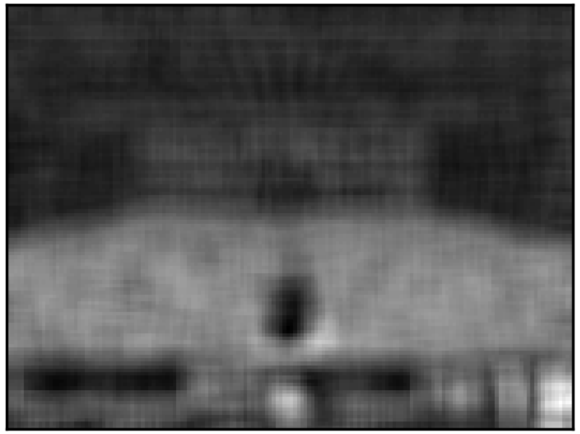


Accelerating LASSO with Reduced Precision

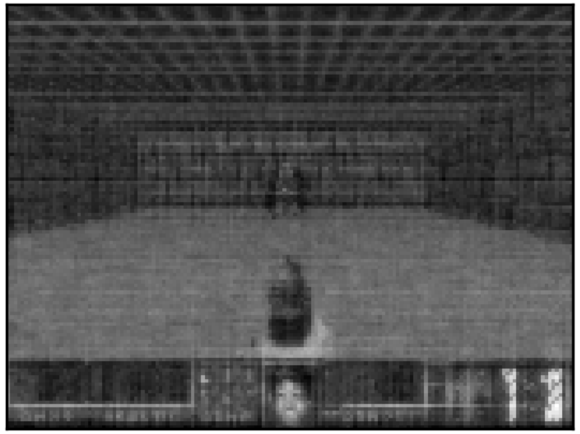


Noisy Frame 120x160 [noise: 0.1]

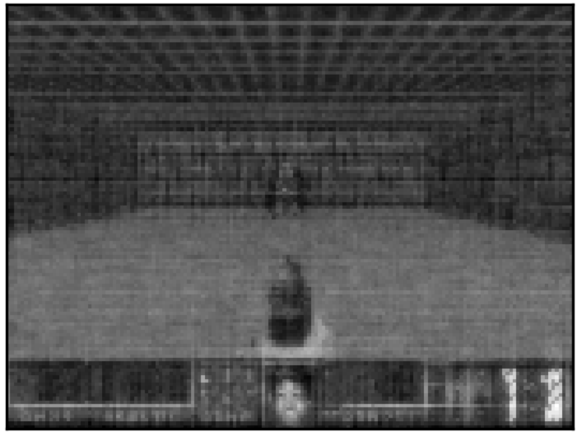
LASSO_GPU--FP16 [6.09±0.22 sec]



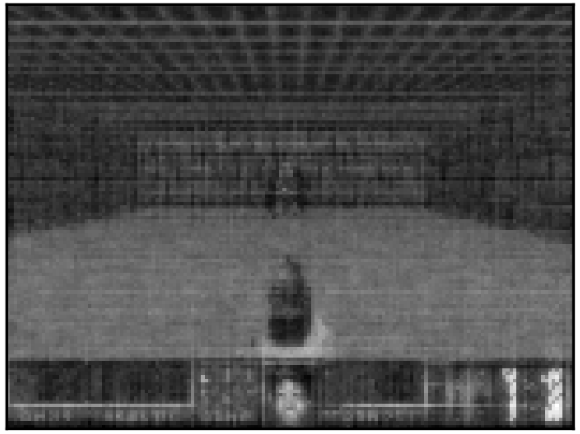
LOWPASS [0.8± 1 msec]



LASSO_CPU--1 ITER [6.2±0.12 sec]



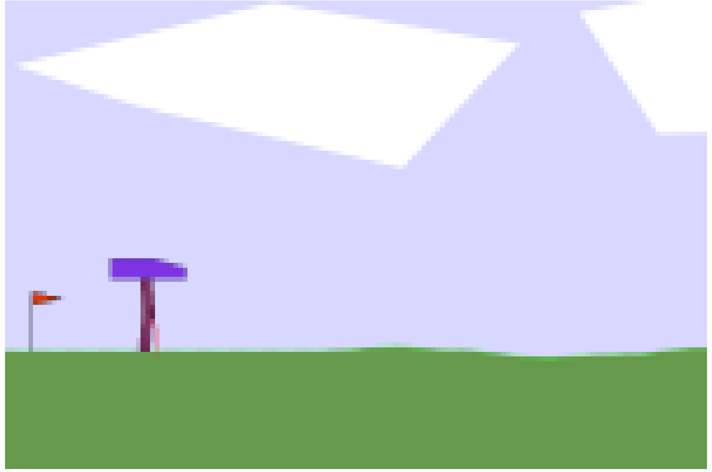
LASSO_CPU--2 ITERS [6.77±0.06 sec]

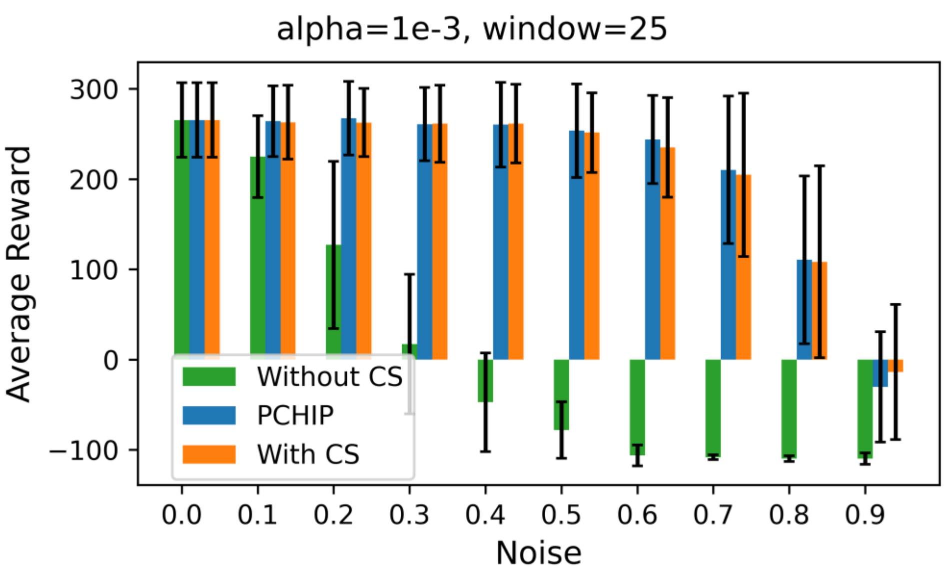


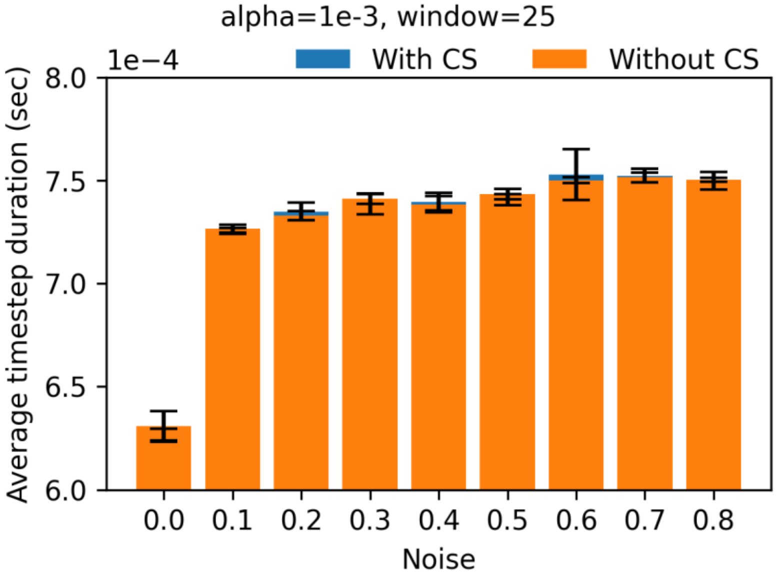
LASSO_CPU--30 ITERS[8.5±0.35 sec]

- By using high parallelization and reduced precision available in GPUs, we can reduce the time for LASSO optimization significantly.

Results: RL + CS for Bipedal Walker







- 4-joint walker robot environment
- In this case, the state inputs are the hull angle speed, angular velocity, horizontal speed, vertical speed, position of joints and joints angular speed, legs contact with ground, and 10 lidar rangefinder measurements.
- Noise is simulated by simply zeroing out some of the sensor values randomly.
- We maintain a history of previous observations for a certain time horizon [25 timesteps] for CS reconstruction.

Conclusion

- Performance of RL agents degrade significantly with observation noise introduced due to missing data.
- We use CS to reconstruct the missing data and recover the performance of RL agents in noisy environments. For sake of example, we use LASSO optimization.
- When input is relatively low dimensional, the computation for CS overhead is minimal.
- When input is high dimensional (e.g., pixels), CS requires large matrix operations. Using GPUs and reduced precision can accelerate the optimization process.

Shaswot SHRESTHAMALI
https://www.acsl.ics.keio.ac.jp/
www.shaswot.com

Masaaki KONDO
https://www.acsl.ics.keio.ac.jp/


Thank You