Memory-Consistent Neural Networks for Imitation Learning

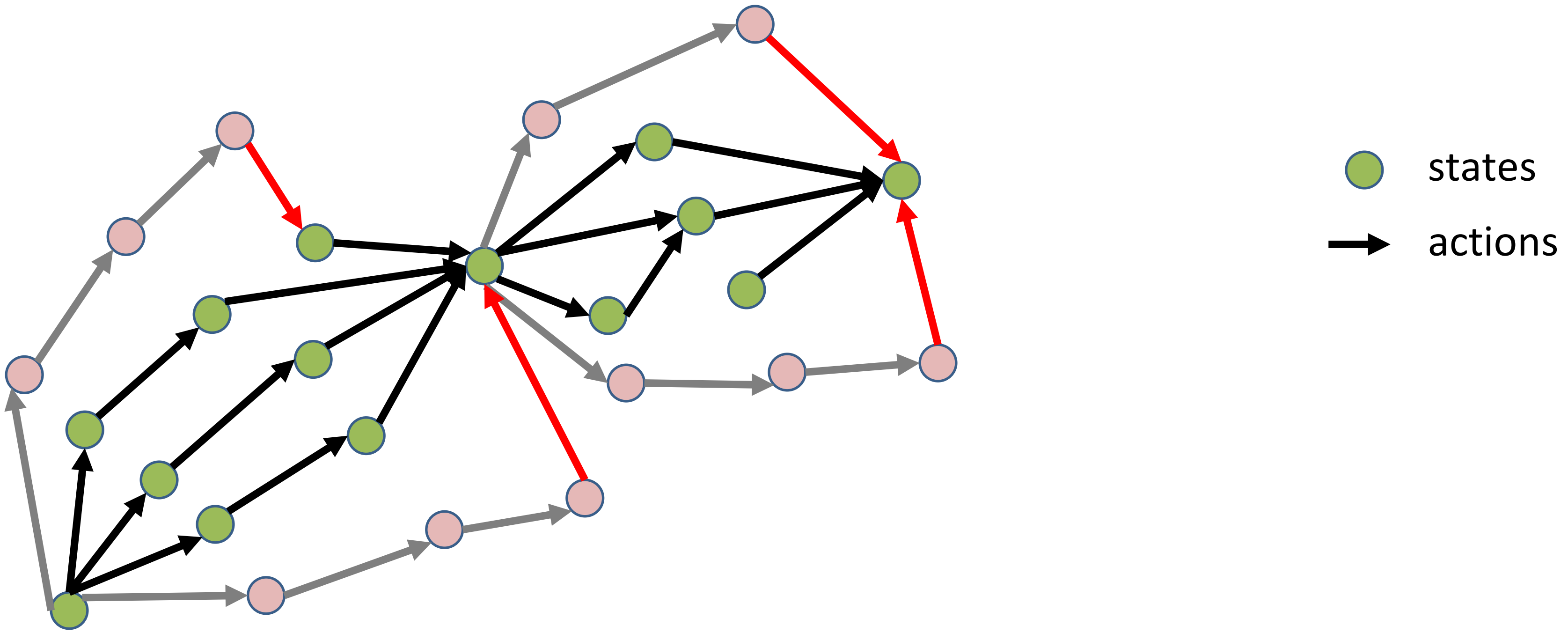
Kaustubh Sridhar, Souradeep Dutta,
Dinesh Jayaraman, James Weimer, Insup Lee

Penn Engineering
University of Pennsylvania

Motivation

Compounding errors in imitation learning on offline datasets: can we avoid them altogether?

Current methods include corrective data from online experience, queryable experts, reward labels, etc.

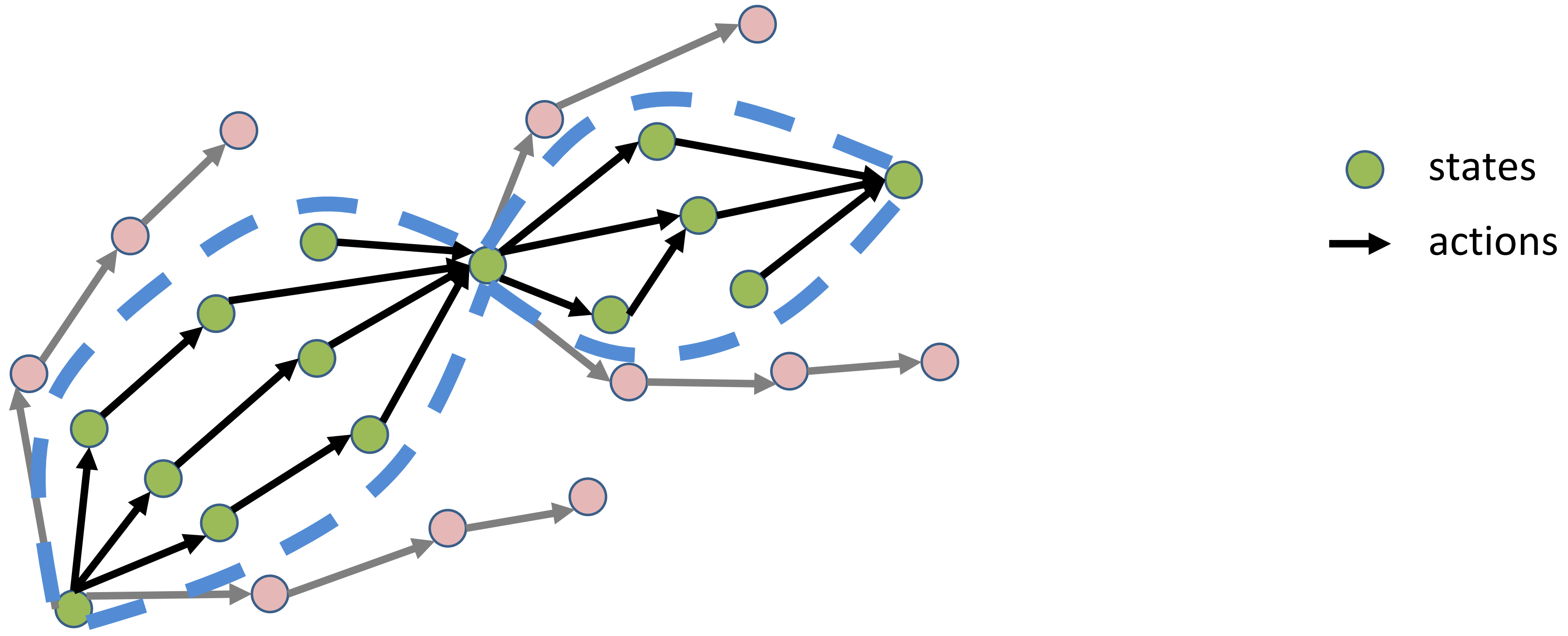


Motivation

Compounding errors in imitation learning on offline datasets: can we avoid them altogether?

Current methods include corrective data from online experience, queryable experts, reward labels, etc.

But what if we could design a constrained model class for simple (scalable) behavior cloning?



Problem Formulation

Unknown expert policy π^*

Demonstrations dataset $D = \{(s_0, a_0), (s_1, a_1), \ldots, (s_N, a_N)\}$

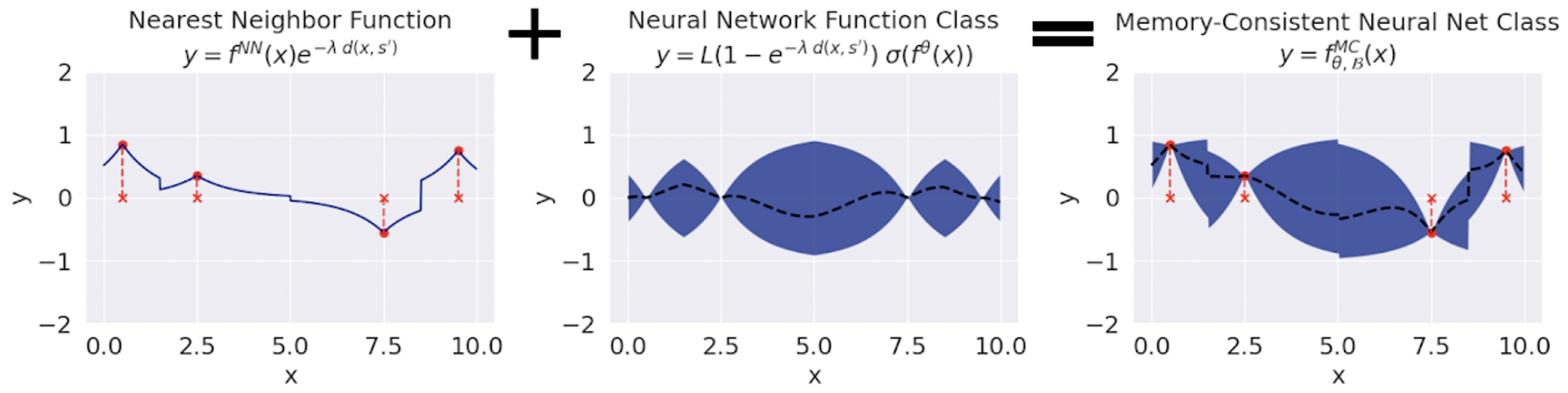
Objective: reduce sub-optimality gap $J(\pi^*) - J(\hat{\pi})$

where $$J(\pi) = \mathbb{E}_\pi \left[\sum_{t=1}^{H} \mathcal{R}(s_t, a_t) \right]$$

Key Idea

New model class for behavior cloning that interpolates between nearest “memory” and vanilla neural net:

Behaves like nearest neighbors near “memories” and vanilla neural net away from “memories”.



Key Idea

New model class for behavior cloning that interpolates between nearest “memory” and vanilla neural net:

Behaves like nearest neighbors near “memories” and vanilla neural net away from “memories”.

Exponential term with rate of change knob

Interpolation in space

Amplitude knob (1 - exponential term)

$$f^{MC}_{\theta,\mathcal{B}}(x) = \underbrace{f^{NN}(x)\left(e^{-\lambda\, d(x,s')}\right)}_{\text{Nearest Memory Neighbour Function}} + \underbrace{L\left(1 - e^{-\lambda\, d(x,s')}\right)\sigma(f^{\theta}(x))}_{\text{Constrained Neural Network Function Class}}$$

Neural net params

Codebook of “memories”

New tanh-like activation.

Training a MCNN

(1) Learn memories

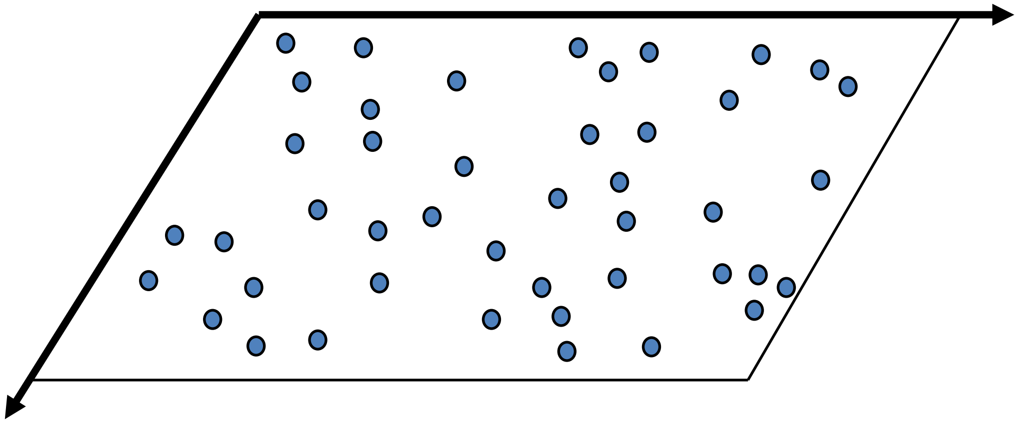
Neural gas clustering algorithm*

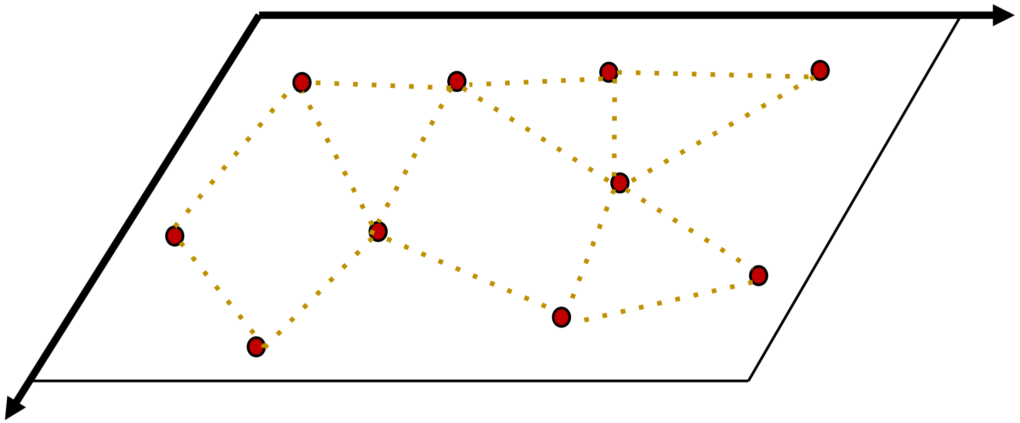
Neural gas

Nearest neighbours of neural gas nodes in dataset

State or (image) embedding space

Codebook of memories (graph or list)





* This clustering technique [Fritzke NeurIPS 1994] produces a graph and hence was (initially) chosen for fast inference. But, brute force search at inference time on GPUs is faster! (cue GPUs go brr meme)

Training a MCNN

(2) Learn MCNN

$$f^{MC}_{\theta,\mathcal{B}}(x) = \underbrace{f^{NN}(x)\left(e^{-\lambda\, d(x,s')}\right)}_{\text{Nearest Memory Neighbour Function}} + \underbrace{L\left(1 - e^{-\lambda\, d(x,s')}\right)\,\sigma(f^{\theta}(x))}_{\text{Constrained Neural Network Function Class}}$$

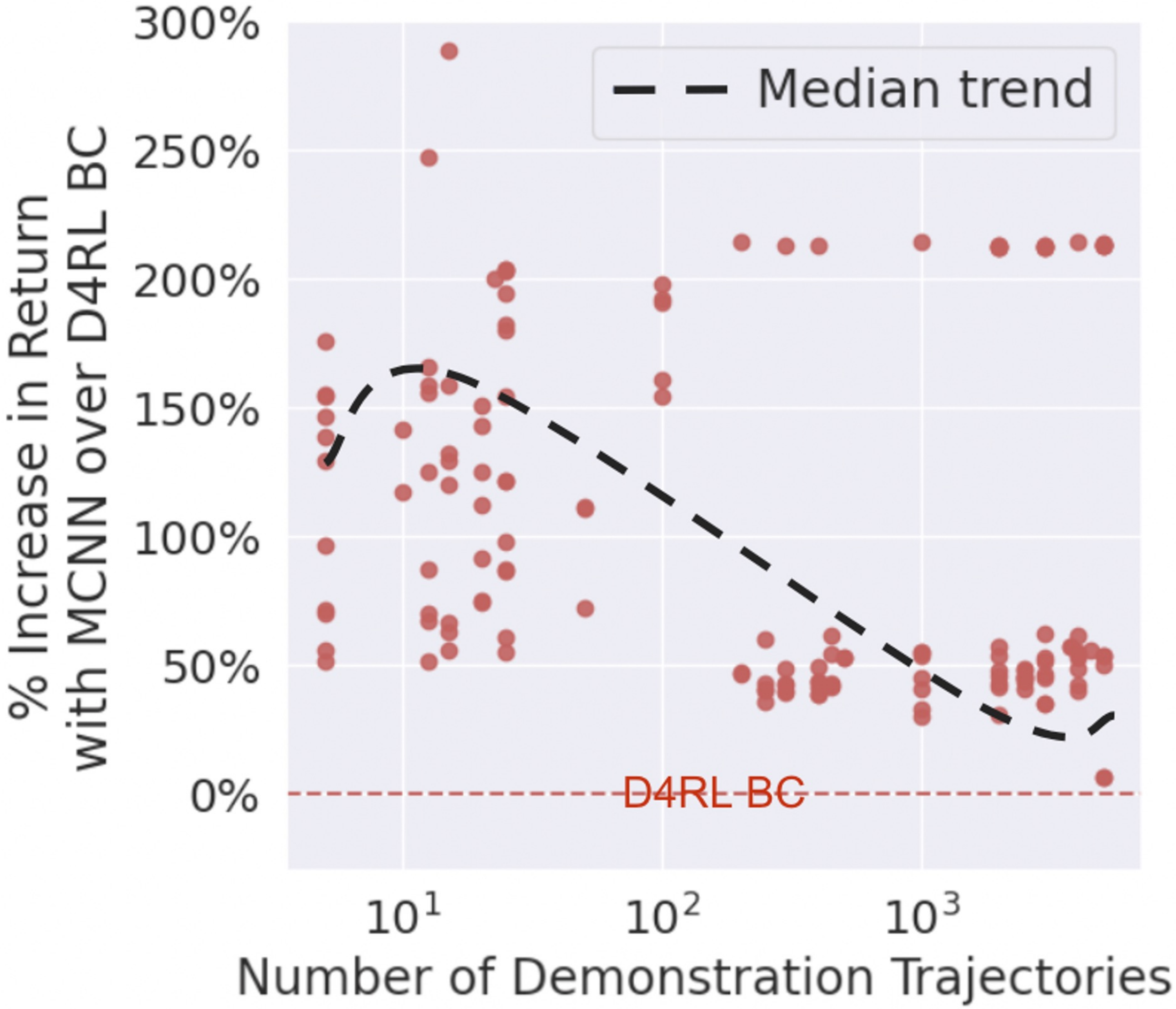
Maps input to nearest memory (retrieval)

Vanilla neural net of input

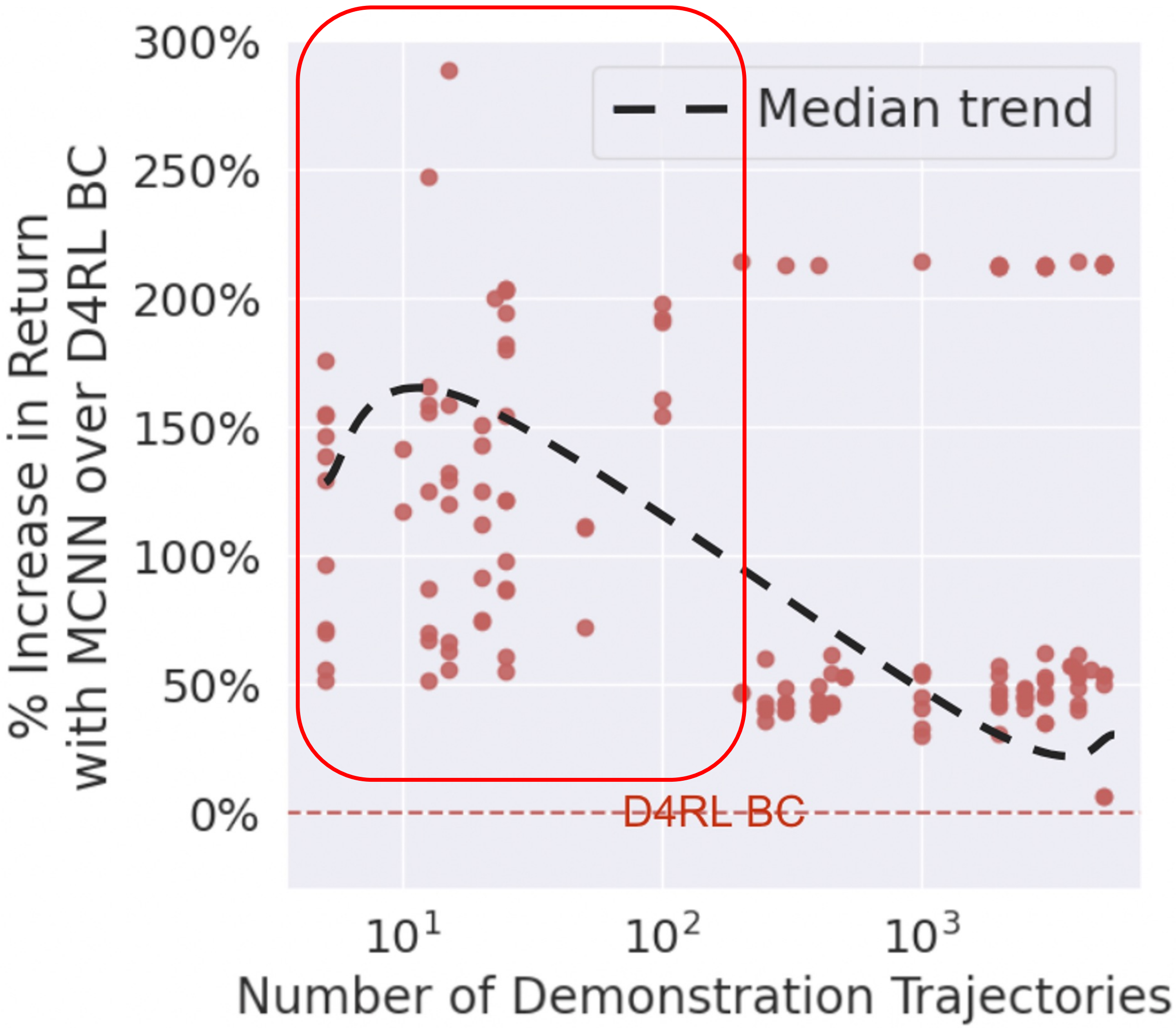
Sample batch B from $\mathcal{D}$.

$$\theta \leftarrow \theta - \nabla\, \mathbb{E}_{(s_i,a_i)\sim B}\, \mathcal{L}(f^{MC}_{\theta,\mathcal{B}}(s_i), a_i)$$

Teaser (for results)



Teaser (for results)



Guarantees

Assumption 4.3 (Realizability). We assume that the expert policy π^* belongs to the function class $\mathfrak{F}$.

Definition 4.4 (Most Isolated State). For a given set of memory points $\mathcal{B}|_S$, we define the most isolated state $s^I_{\mathcal{B}|_S} := \arg\max_{s \in S} \left(\min_{m \in \mathcal{B}|_S} d(s, m) \right)$, and consequently the distance of the most isolated point as $d^I_{\mathcal{B}|_S} = \min_{m \in \mathcal{B}|_S} d(s^I_{\mathcal{B}|_S}, m)$

Theorem 4.7. *The sub-optimality gap* $J(\pi^*) - J(\hat{\pi}) \leq min\{H, H^2|\mathcal{A}|L\left(1 - e^{-\lambda\, d^I_{\mathcal{B}|_S}}\right)\}$

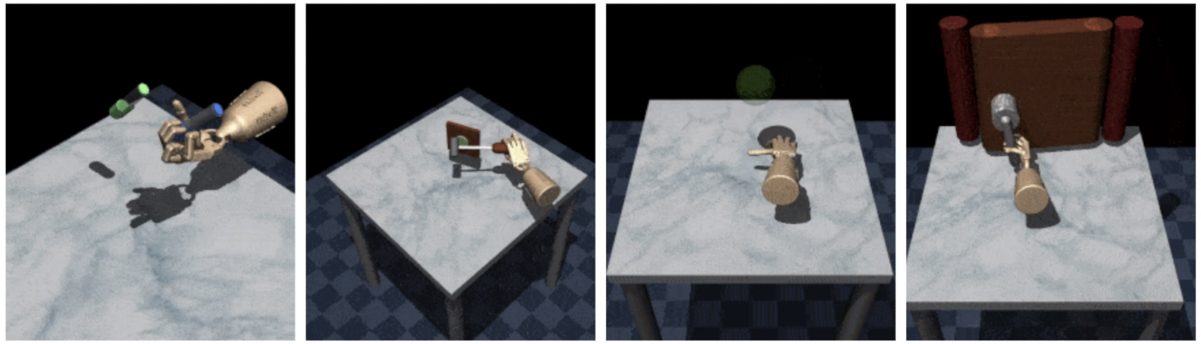
Corollary 4.8. *if* $\mathcal{B}_i \subseteq \mathcal{B}_j$ *and* $H \geq H^2|\mathcal{A}|L\left(1 - e^{-\lambda\, d^I_{\mathcal{B}|_S}}\right)$

$\Rightarrow$ *lower performance gap*

"more (widespread) memories lead to a lower performance gap (upto a limit)"

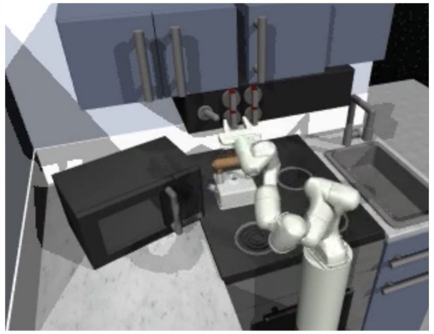
Results

Datasets and Envs:



Adroit (D4RL) Pen, Hammer, Relocate, and Door
[Proprioception, 24-30 dim action spaces,
5k (human) to 1M (expert) transitions]

D4RL CARLA
[Images, 2D actions,
100k (expert)
transitions]

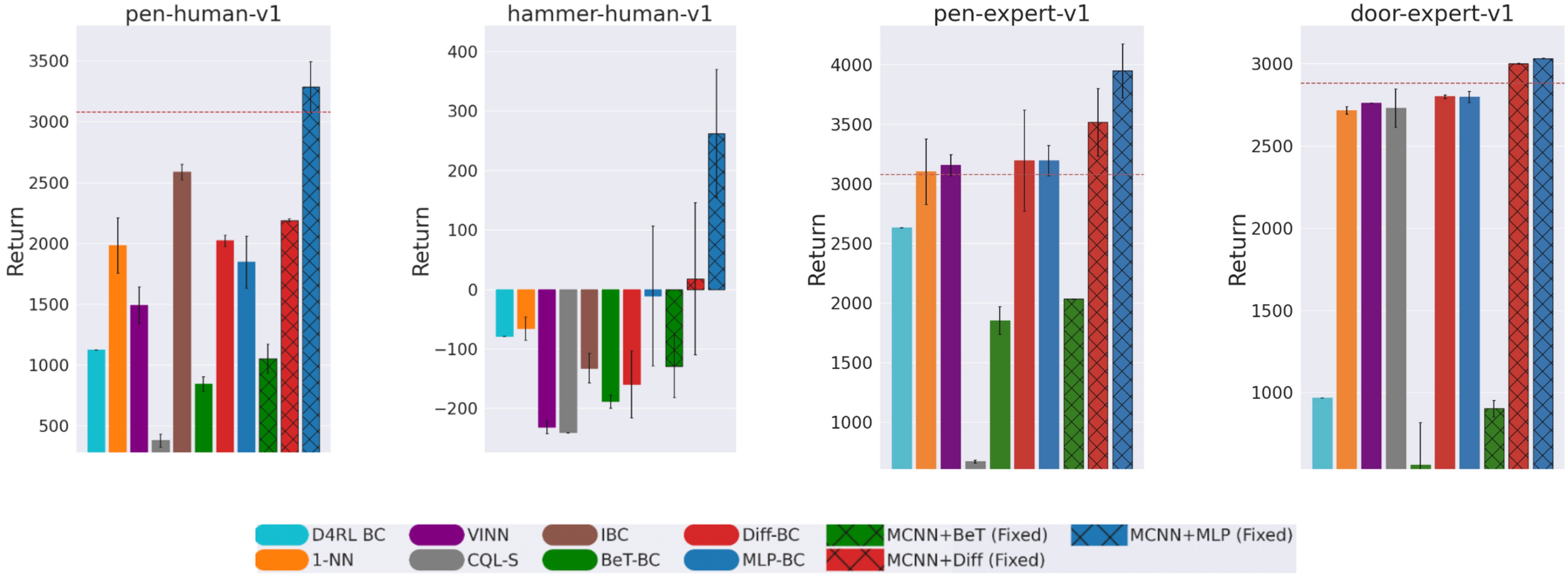


Franka Kitchen
[Proprioception,
high multimodality,
130k (human)
transitions]

Architecture and loss:

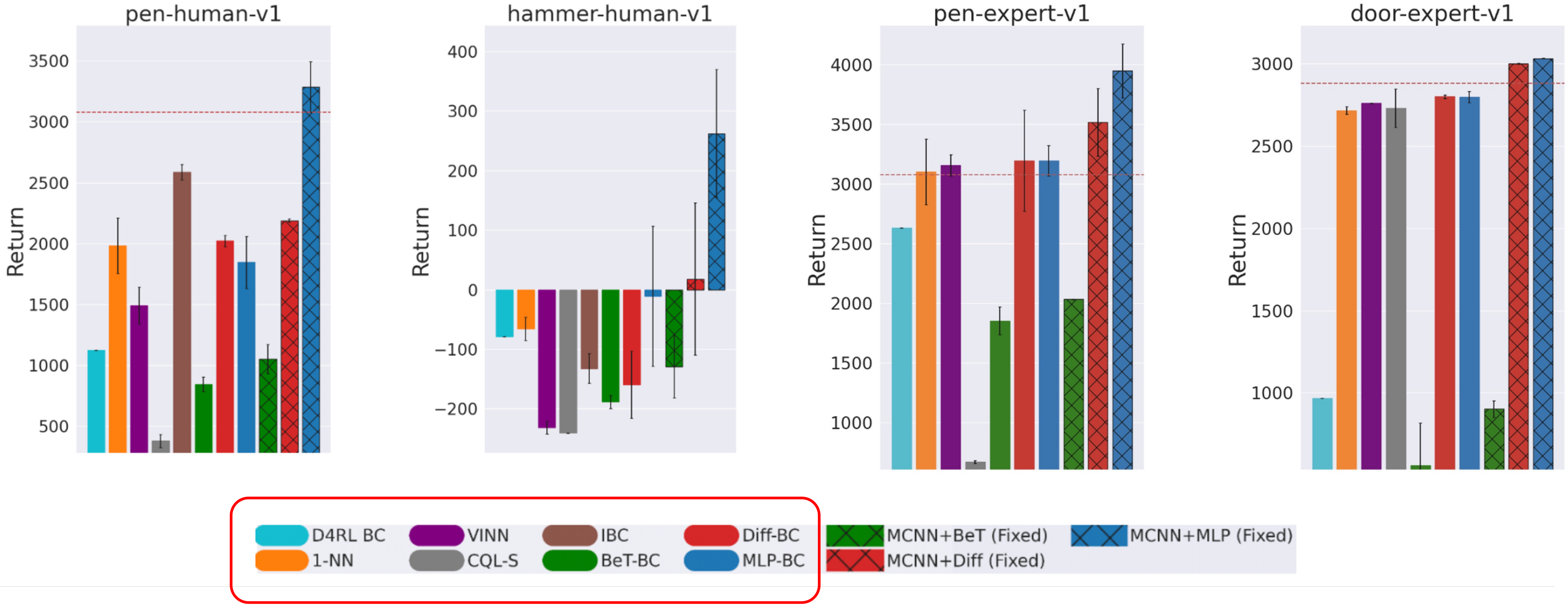
1. **MCNN-MLP** with **Mean Squared Error**
2. **MCNN-Diffusion** with a **Denoising Diffusion Process** and **Mean Squared Error** to predict noise/result at every step
3. **MCNN-Behavior Transformer (BeT)** with predicting **Action Buckets (via Cross Entropy Loss)** and **Action Offsets (via Mean Squared Error)**
4. **Embedding images** with an off-the-shelf ResNet 34 encoder

Results

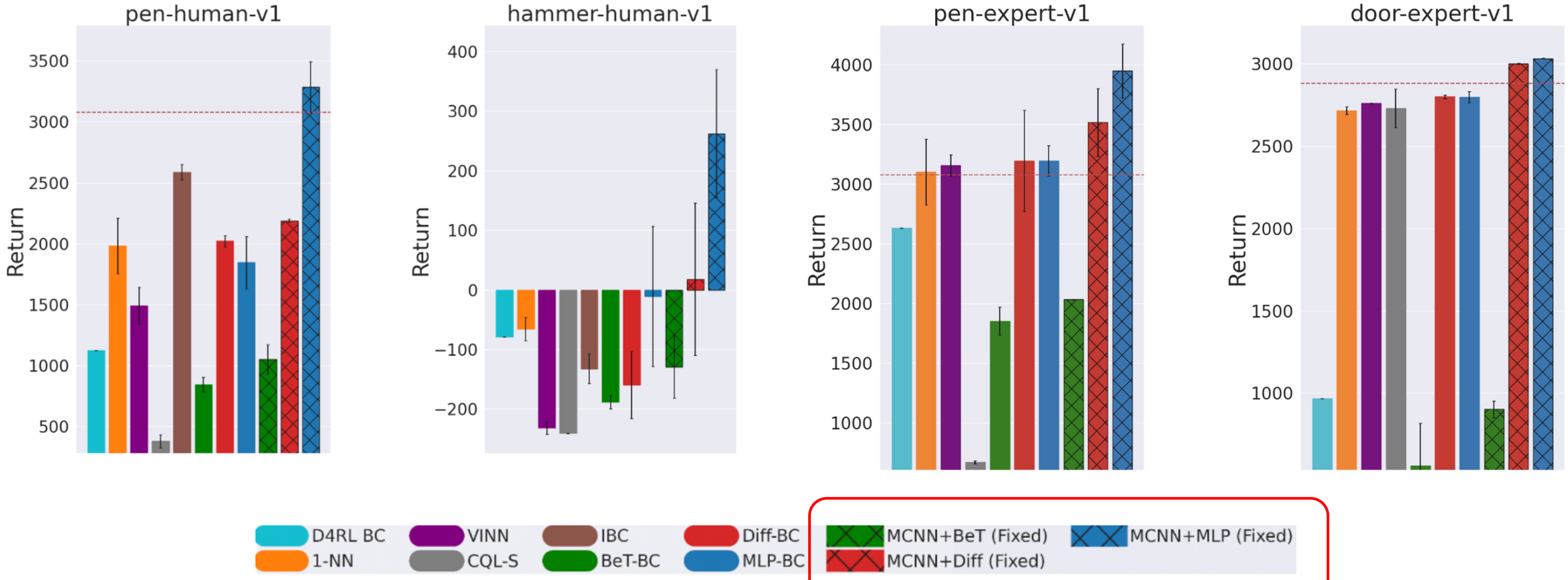


pen-human-v1
hammer-human-v1
pen-expert-v1
door-expert-v1
Return
D4RL BC
1-NN
VINN
CQL-S
IBC
BeT-BC
Diff-BC
MLP-BC
MCNN+BeT (Fixed)
MCNN+Diff (Fixed)
MCNN+MLP (Fixed)

Results

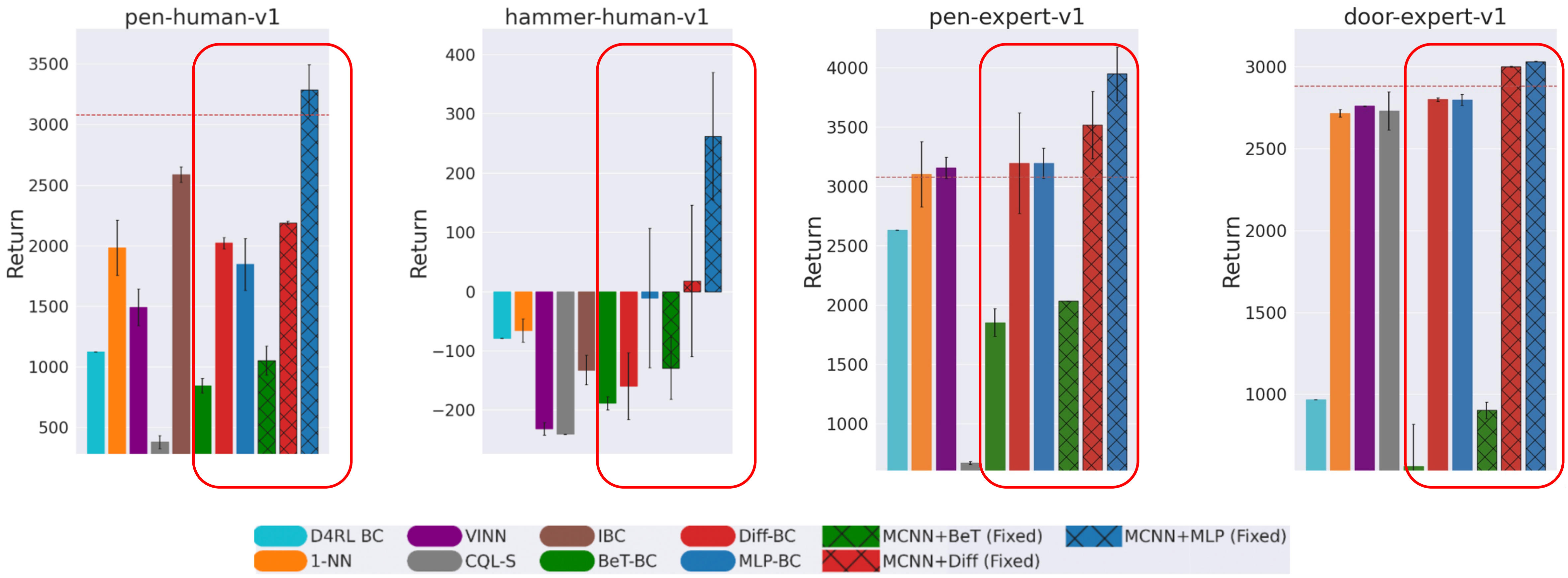

pen-human-v1
hammer-human-v1
pen-expert-v1
door-expert-v1
Return
D4RL BC
1-NN
VINN
CQL-S
IBC
BeT-BC
Diff-BC
MLP-BC
MCNN+BeT (Fixed)
MCNN+Diff (Fixed)
MCNN+MLP (Fixed)

Results



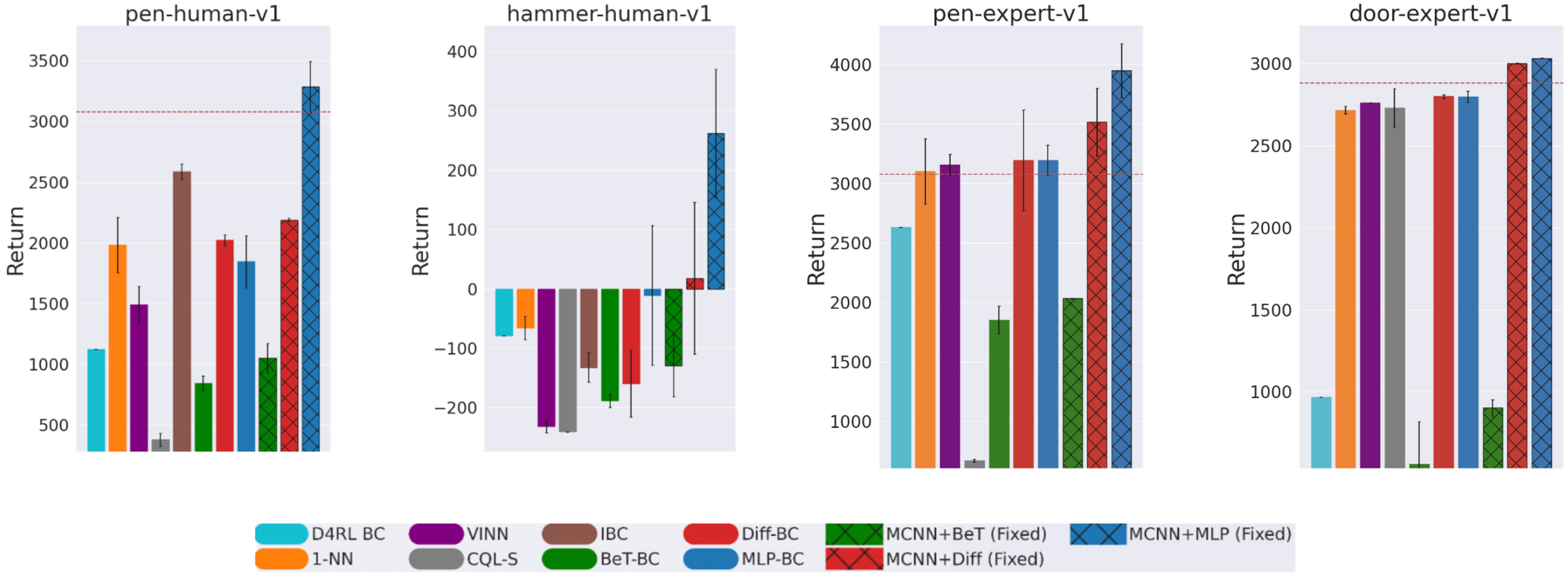
pen-human-v1
hammer-human-v1
pen-expert-v1
door-expert-v1
Return
3500
3000
2500
2000
1500
1000
500
400
300
200
100
0
−100
−200
4000
D4RL BC
1-NN
VINN
CQL-S
IBC
BeT-BC
Diff-BC
MLP-BC
MCNN+BeT (Fixed)
MCNN+Diff (Fixed)
MCNN+MLP (Fixed)

Results



pen-human-v1
hammer-human-v1
pen-expert-v1
door-expert-v1
Return
3500
3000
2500
2000
1500
1000
500
400
300
200
100
0
−100
−200
4000
D4RL BC
1-NN
VINN
CQL-S
IBC
BeT-BC
Diff-BC
MLP-BC
MCNN+BeT (Fixed)
MCNN+Diff (Fixed)
MCNN+MLP (Fixed)

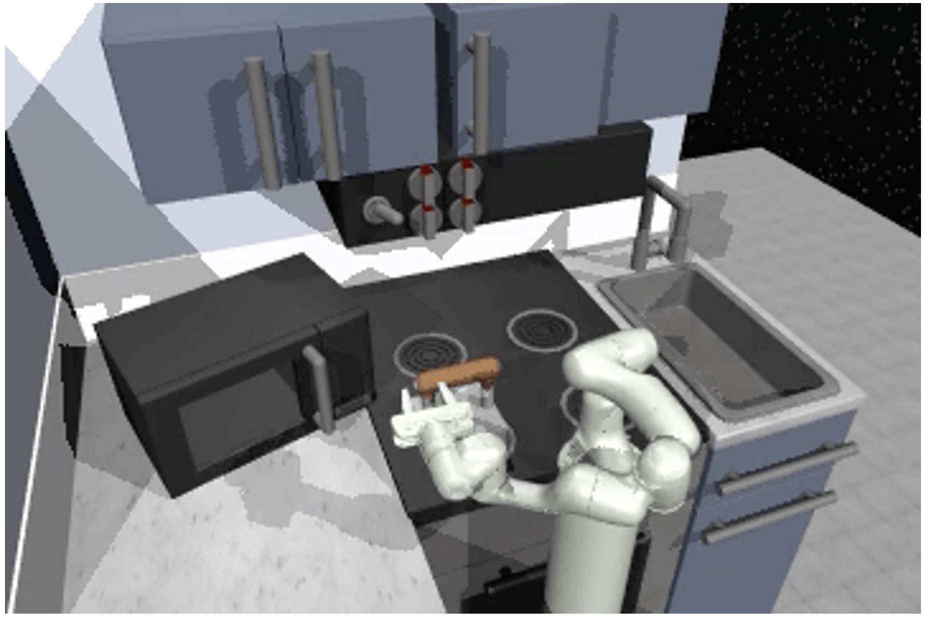
Results

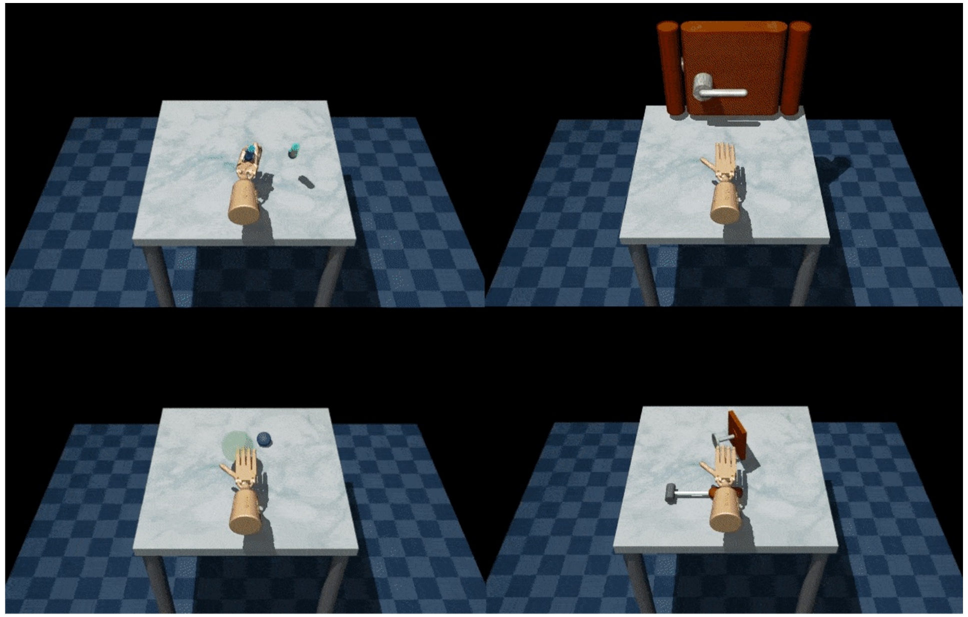


with similar significant improvements in other Adroit tasks, CARLA, Franka Kitchen.

Key Takeaways

1. MCNN can help with generalization to test envs from small datasets with little hyperparameter tuning.
2. Broadly, semi-parametric methods may hold the key to generalization in robotics/embodied AI (already a key part of the RAG+LLM world).





More information can be found at …

bit.ly/mcnn