

# Raising Body Ownership in End-to-End Visuomotor Policy Learning via Robot-Centric Pooling

Zheyu Zhuang<sup>1</sup> Ville Kyrki<sup>2</sup> Danica Kragic<sup>1</sup>

**Abstract**—We present Robot-centric Pooling (RcP), a novel pooling method designed to enhance end-to-end visuomotor policies by enabling differentiation between the robots and similar entities or their surroundings. Given an image-proprioception pair, RcP guides the aggregation of image features by highlighting image regions correlating with the robot’s proprioceptive states, thereby extracting robot-centric image representations for policy learning. Leveraging contrastive learning techniques, RcP integrates seamlessly with existing visuomotor policy learning frameworks and is trained jointly with the policy using the same dataset, requiring no extra data collection involving self-distractors. We evaluate the proposed method with reaching tasks in both simulated and real-world settings. The results demonstrate that RcP significantly enhances the policies’ robustness against various unseen distractors, including self-distractors, positioned at different locations. Additionally, the inherent robot-centric characteristic of RcP enables the learnt policy to be far more resilient to aggressive pixel shifts compared to the baselines. Code available at: <https://github.com/Zheyu-Zhuang/RcP>

## I. INTRODUCTION

Body ownership enables us to differentiate our own body from objects in our surroundings (*self-recognition*) and from other individuals (*self-other distinction*) [1]. These aspects are crucial for robots, especially in shared or multi-robot settings, where a robot’s actions should remain unaffected by environmental objects and other robots. Acknowledging that body ownership in humans involves complex multisensory integration and cognitive processes [2], and drawing on prior work [3], [4], [5], we describe body ownership for robots at the visuomotor level, focusing on simulating features of self-recognition and self-other distinction.

In this work, we introduce Robot-centric Pooling (RcP) to address severely limited self-recognition and self-other distinction capability in conventional end-to-end visuomotor policy learning. This novel pooling method explicitly integrates both visual information and the robot’s proprioceptive state, setting it apart from traditional pooling methods that rely solely on image data. RcP computes alignment scores between image regions and the robot’s proprioceptive states to derive image representations that reflect a robot-centric perspective (Fig.1). Notably, RcP is fully self-supervised and task-agnostic, allowing it to integrate seamlessly with existing standard CNN-based visuomotor policy learning

<sup>1</sup>The authors are with the Robotics, Perception and Learning Lab, EECS, at KTH Royal Institute of Technology, Stockholm, Sweden [zheyuzh, dani@kth.se](mailto:zheyuzh, dani@kth.se)

<sup>2</sup>The authors are with the Intelligent Robotics Group, Department of Electrical Engineering and Automation (EEA), Aalto University, Espoo, Finland. [ville.kyrki@aalto.fi](mailto:ville.kyrki@aalto.fi)

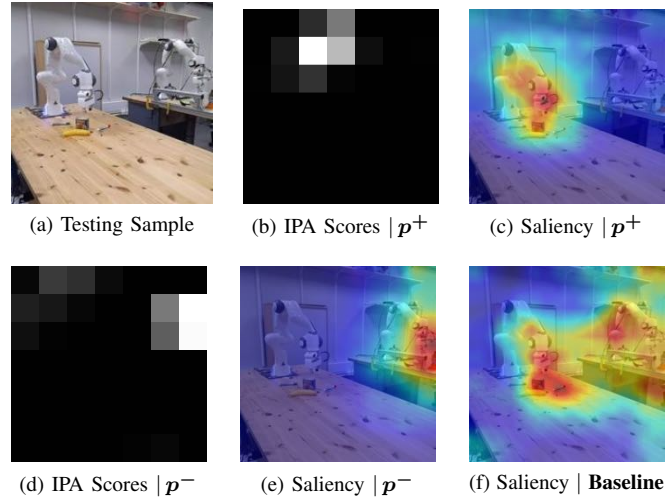


Fig. 1: **Body ownership via Robot-centric Pooling.** RcP enables a conventional policy regression baseline to foster self-recognition and the ability to distinguish self from others. (a): A testing sample including a self-distractor (right). (b): Image-proprioception alignment scores for self-state,  $p^+$ . (c): Image saliency map [6] with  $p^+$  based on regressed policy (warmer colours indicate higher relevance). (d): IPA scores for the distractor’s state  $p^-$ . (e): Saliency map with  $p^-$ . (f): Saliency map from the Spatial-Softmax [7] baseline.

frameworks. It can be jointly trained with the policy using the same training data, requiring no additional data collection.

We evaluate RcP with reaching tasks, in both simulated and real-world settings. Our experimental results show that:

- Conventional end-to-end learning baselines exhibit considerable sensitivity to environmental distractions and self-distractors, revealing a fundamental deficiency in the development of body ownership.
- Robot-centric Pooling (RcP) demonstrates significant enhancement against distractions, showing only a slight decrease in success rates (from 96% to 92%) amidst a self-distractor in real-world experiments, as opposed to baseline models which plummet to below 15%.
- Benefiting from the robot-centric nature, RcP significantly enhances policy robustness against aggressive image shifts compared to baseline methods.

To the best of our knowledge, this is the first demonstration of both the self-other distinction and the self-recognition capabilities of body ownership in the context of end-to-end visuomotor policy learning.

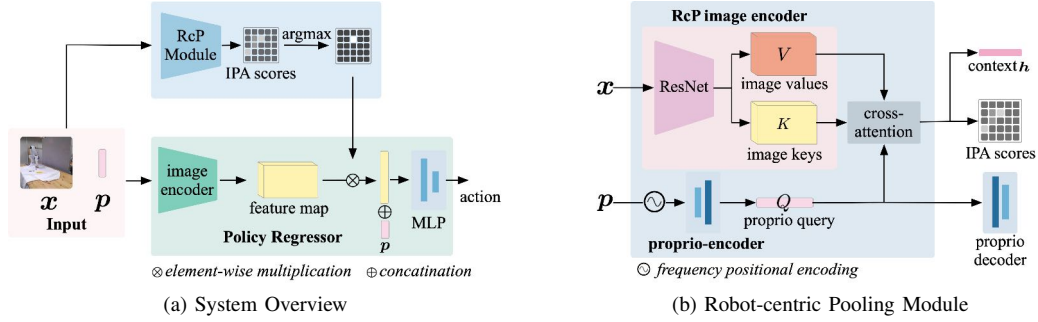


Fig. 2: **System Overview and the Robot-centric Pooling Module.** (a): Robot-centric Pooling extracts the most relevant feature corresponding to the identified self for the regression task. (b): RcP computes Image-Proprioception Alignment (IPA) scores from an image-proprioseption pair  $(x, p)$  and aggregates image values accordingly to create a context vector for contrastive learning and image representation in the regression pipeline.

## II. RELATED WORK

**Robot self-recognition and self-other distinction** capabilities are studied through both non-learning and learning-based methods. Non-learning-based approaches involve correlating observed motion with robot actions using techniques such as mutual information [8], [4], dense image Jacobian estimation [5], or temporal delay [9]. Learning-based approaches are studied at both intermediate visual cue level, such as optical flow [3] and pixel level [10], [11], [12]. The training data of the learning-based approaches are self-labelled by associating the motor inputs with the observations. However, the development of self-recognition and self-other distinction through end-to-end visuomotor policy learning and their impact on the policies remains unexplored.

**Image-proprioseption integration** is fundamental for end-to-end visuomotor policy learning in robotics and plays a vital role in cultivating body ownership in humans. While the specific integration mechanism in human cognition remains elusive, robotics leverage the *concatenation paradigm* for this purpose. Here, image representations are extracted from either single or sequential observations using CNNs [7], [13], [14], or transformers [15], [16]. Simultaneously, the proprioceptive states, which may include manipulator joint positions, velocities, and end-effector poses, are directly read from the manipulator [15], [14], or further encoded through a multi-layer perceptron [17]. The two representations are then concatenated for the downstream task. Despite this integration, end-to-end learning approaches have yet to demonstrate an innate development of self-recognition and self-other distinction capabilities.

**Contrastive learning** techniques form the cornerstone of RcP for tackling the challenge of developing body ownership with existing single-manipulator datasets. These techniques [18], [19], [20], [21] employ discriminative learning objectives to encourage similarities within positive pairs and dissimilarities within negative pairs. The generation of these pairs are typically constructed through sequences of data augmentation techniques [19], [20], [22]. A typical training process involves dual encoders to process two sets of separately augmented samples [19], [23], [24]. In this work,

we adopt the Momentum Contrastive Learning framework (MOCO) [21] proposed by He et al., a strategy where the second encoder’s weights are updated as a momentum moving average of the first’s, and maintaining a negative sample queue for past keys to increase the negative sample size.

## III. METHODOLOGY

### A. The Robot-centric Pooling Module

As depicted in Fig. 2b, given an image-proprioseption pair  $(x, p)$ , the RcP module uses a cross-attention mechanism [25] to align the image and proprioceptive latent representations while enabling self-supervised contrastive learning. It outputs a context vector  $h$  for contrastive learning and Image-Proprioception Alignment (IPA) scores. The IPA scores are then transformed into a binary mask, identifying the most relevant feature for the regression task (Fig. 2a).

The RcP image encoder  $f_{\theta}$ , with learnable parameters  $\theta$ , encodes an input image  $x$  into image keys  $k$  and values  $v$ :

$$(k, v) = f_{\theta}(x). \quad (1)$$

Specifically, the RcP image encoder’s backbone is a modified ResNet18 [26] with the last average pooling and the classification layers removed. Two learnable projection matrices, denoted as  $W_k$ ,  $W_v$ , project the layer-normalised [27] features into image keys  $k$  and values  $v$ :

$$k = W_k (\text{LN}(\text{ResNet}_{\psi}(x))) \in \mathbb{R}^{(hw) \times d}, \quad (2)$$

$$v = W_v (\text{LN}(\text{ResNet}_{\psi}(x))) \in \mathbb{R}^{(hw) \times d}. \quad (3)$$

Here,  $\psi$  denotes the learnable parameters,  $\text{LN}(\cdot)$  denotes the LayerNorm operation,  $hw$  represents the product of the spatial dimensions, and  $d$  specifies the feature dimension.

Frequency encoding has shown advantages in mapping continuous coordinate inputs to higher dimensions, which facilitates better approximation of high-frequency functions [28]. Since robot proprioceptive states  $p$  are continuous and periodic (e.g., joint angles), they are first encoded into a frequency representation, formally defined as:

$$\gamma(p) = \bigoplus_{l=0}^{L-1} (\sin(2^l \pi p), \cos(2^l \pi p)), \quad (4)$$

where  $\oplus$  denotes the concatenation of all frequency levels from 0 to  $L - 1$ . The resulting representation  $\gamma(\mathbf{p})$  is then processed through a two-layer perceptron  $Q(\cdot)$ , yielding a proprioception query:

$$\mathbf{q} := Q(\gamma(\mathbf{p})) \in \mathbb{R}^{1 \times d}. \quad (5)$$

A proprioception decoder with mirrored architecture as  $Q(\cdot)$  facilitates the learning of the proprioception query, forcing meaningful latent representation of the robot state.

Denote the IPA scoring map as  $s \in \mathbb{R}^{h \times w}$ , and the vectorisation operator as  $\text{vec} : \mathbb{R}^{h \times w} \rightarrow \mathbb{R}^{1 \times (hw)}$ . Based on the image keys  $\mathbf{k}$  and the proprioception query  $\mathbf{q}$ , the IPA scores are computed as their cosine similarities:

$$\text{vec}(\mathbf{s}) = \text{softmax}(\mathbf{q}\mathbf{k}^\top) \quad (6)$$

Finally, the image values  $\mathbf{v} \in \mathbb{R}^{(hw) \times d}$  are aggregated based on the IPA score vector to form the context vector  $\mathbf{h}$  following the cross-attention formulation:

$$\mathbf{h} = \text{softmax}(\mathbf{q}\mathbf{k}^\top) \mathbf{v}. \quad (7)$$

In practice, the IPA scores between self-distractors that resemble the robot's state can be similar, introducing noise into the policy regression pipeline. To mitigate this, an  $\text{argmax}$  operation is applied to the IPA scoring map, generating a binary mask that highlights only the image region most relevant to the robot's state:

$$\text{RcP}(\mathbf{x}, \mathbf{p}) := \mathbf{1}_{ij} \left( \underset{i,j}{\text{argmax}} s_{ij} \right), \quad (8)$$

where the  $\text{argmax}$  operation identifies the indices  $(i, j)$  corresponding to the maximum value in the IPA scoring map  $s$ , and  $\mathbf{1}_{ij}(\cdot)$  represents the indicator function that assigns 1 to the location  $(i, j)$  and 0 elsewhere, resembling a global max-pooling operation with spatial locations specified by the maximum IPA score (Eq. (10)).

### B. Emergence of Body Ownership via Contrastive Learning

The image keys corresponding to the 'self' should achieve high IPA scores in comparison to those associated with the environment or other bodies. Therefore, context features derived from the same proprioceptive query are expected to be similar, despite changes in background or self-distractors. While context features extracted from different queries should show clear dissimilarity.

RcP achieves this core capability through contrastive learning techniques in a self-supervised fashion. Firstly, a stochastic data augmentation module transforms an image  $\mathbf{x}$  into an anchor image  $\tilde{\mathbf{x}}$  and the corresponding positive image  $\tilde{\mathbf{x}}^+$ . More specifically, as shown in Fig. 3b, this process involves cropping the manipulator region from the image using a roughly calibrated camera and then pasting this crop onto two randomly selected backgrounds at varied spatial locations. Meanwhile, a self-distractor is cropped from a random image within the training dataset and pasted onto one of the two augmented images in a location that ensures the majority of the already pasted robot remains visible. Random scaling is applied to the crops and colour jittering as in [13] is

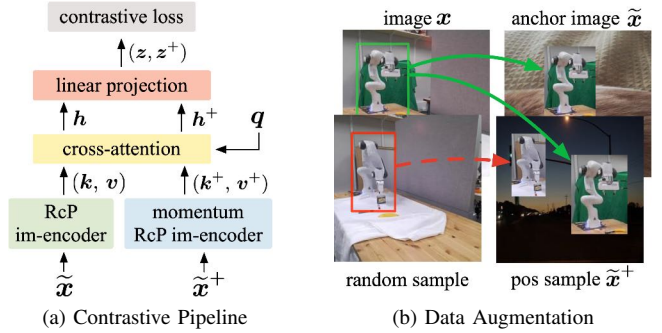


Fig. 3: **Illustration of the Contrastive Learning Framework.** (a): Similar to the pipeline proposed in MOCO [21], the augmented images are separately encoded by the RcP's image encoder and its momentum averaging copy. (b): For each image, the manipulator region is cropped and pasted onto two random backgrounds at random spatial locations (green firm arrows). A self-distractor is cropped from a random image drawn from the training dataset and randomly pasted onto one of the augmented images (red dashed arrow).

applied to the composed images. Each of the two augmented images has an equal probability of being designated as either the anchor  $\tilde{\mathbf{x}}$  or the positive image  $\tilde{\mathbf{x}}^+$ .

As depicted in Fig. 3a, similarly to MOCO [21], we employ two copies of the RcP image encoder (Eq. (1)) for contrastive learning. Note that, the proprioception encoder and decoder are not copied. The first copy  $f_\theta$  receives the gradient updates and the other  $f_{\bar{\theta}}$ , referred to as the momentum image encoder, having its weights updated as a moving-average of  $\theta$ :

$$\bar{\theta} \leftarrow m\bar{\theta} + (1 - m)\theta, \text{ where } m \in [0, 1].$$

Following Eq. (2), these encoders separately process the anchor  $\tilde{\mathbf{x}}$  and the positive image  $\tilde{\mathbf{x}}^+$  into the corresponding keys and values. Meanwhile, the proprioception state  $\mathbf{p}$  is encoded into the proprioception query  $\mathbf{q}$  via Eq. (5). The anchor and positive image-derived keys and values separately perform the cross-attention (as in Eq. (6)) operation with  $\mathbf{q}$ , producing the corresponding context vectors  $\mathbf{h}$  and  $\mathbf{h}^+$ .

The context vector  $\mathbf{h}^+$  encoded by the momentum copy updates the negative samples queue for the next training step. Both context vectors undergo a shared linear projection before computing the contrastive objective. Denote the learnable linear projection weights as  $W \in \mathbb{R}^{d \times d}$ , the context vector  $\mathbf{h}$  after the linear projection is defined as  $\mathbf{z} := W\mathbf{h}$ . We use InfoNCE Loss [18] as the contrastive objective. Given  $K$  projected context vectors saved in the negative sample queue,  $\{\mathbf{z}_0^-, \dots, \mathbf{z}_K^-\}$ , and the positive pair  $(\mathbf{z}, \mathbf{z}^+)$ , the objective is expressed as:

$$\mathcal{L}_{\text{moco}} = -\log \frac{\exp(\mathbf{z} \cdot \mathbf{z}^+ / \tau)}{\sum_{i=0}^K \exp(\mathbf{z} \cdot \mathbf{z}_i^- / \tau)}, \quad (9)$$

where  $\tau$  is a scalar temperature hyper-parameter.

### C. Policy Regression

The image encoder for the policy regressor is also a modified version of ResNet18, with the last classification and average pooling layers removed. Two additional convolution layers with  $3 \times 3$  kernels, denoted as  $\text{Conv}(\cdot)$ , are added to enlarge the receptive field of extracted features. This is critical to compensate for the receptive field reduction caused by the  $\text{argmax}$  operation within RcP, ensuring that the receptive field of features at each spatial location adequately covers the entire input image. The image feature after Robot-centric Pooling,  $\mathbf{g} \in \mathbb{R}^d$ , is then expressed as

$$\mathbf{g} = \text{RcP}(\mathbf{x}, \mathbf{p}) \otimes \text{Conv}(\text{ResNet}_{\phi}(\mathbf{x})), \quad (10)$$

where  $\phi$  denotes the learnable parameters within the image backbone,  $\text{RcP}(\cdot)$  represents the Robot-centric Pooling operation as defined in Eq. (8), and  $\otimes$  denotes the element-wise multiplication broadcasting over the feature dimension. The image feature  $\mathbf{g}$  is then concatenated with the proprioceptive state  $\mathbf{p}$  to form the input vector for policy regression.

The policy regressor  $\pi_{\xi}(\cdot)$  is a two-layer perceptron. We use the L1 loss as the regression objective. Given an observation  $(\mathbf{x}, \mathbf{p})$  pair and the corresponding ground truth action  $\mathbf{a}$ , the policy regression loss is

$$\mathcal{L}_{\text{policy}} = |(\pi_{\xi}(\mathbf{g}, \mathbf{p}) - \mathbf{a})|. \quad (11)$$

The Robot-centric Pooling module can be pre-trained or trained jointly with the regression pipeline. When jointly trained, a preferential weighting  $\lambda \in [0, 1]$  is applied to balance the gradients between the contrastive loss and the regression loss:

$$\mathcal{L} = \mathcal{L}_{\text{policy}} + \lambda(L_{\text{moco}} + \mathcal{L}_{\text{recon}}), \quad (12)$$

where  $\mathcal{L}_{\text{recon}}$  is a mean square error loss for reconstructing the encoded proprioception query.

## IV. EXPERIMENTS

To evaluate the effectiveness of Robot-centric Pooling (RcP) in fostering body ownership, we conduct a series of reaching experiments in both simulated and real-world environments, intentionally introducing distractions from the environment and other robot entities. In all scenarios, we use second-person camera configurations. This increases the system’s susceptibility to distractors, providing us with a unique opportunity to assess the importance of body ownership in enhancing the robustness of the policies.

### A. Baselines and training details

We select two standard Resnet18-based behaviour cloning networks as the baselines, distinguished by their pooling and pre-training methods. One is pre-trained on ImageNet1K [29], with Spatial-Softmax (SSM) pooling [7]. Unlike average pooling, Spatial-Softmax focuses on the 2D spatial locations of the highest activations within image feature maps. It is used as one of the vision baselines in robomimic [13] and the image-backbone in diffusion policy [14]. The second baseline, R3M [30], benefits

from pre-training on the first-person human activity dataset Ego4D [31] through a contrastive learning approach. While keeping the average pooling layer, R3M is reported to achieve more generalisable feature extraction for manipulation tasks.

Both baseline networks and the regression pipeline of our proposed method follow the same architectural framework as depicted in Fig. 2a. All networks are trained using the AdamW optimiser [32], with a learning rate of  $1e^{-4}$ , weight decay of  $1e^{-6}$ , and a mini-batch size of 128, over the course of 150 epochs. Input images are resized to  $224 \times 224$  pixels and normalised according to the corresponding pre-training scheme utilised. Random colour jittering and pixel shift (up to 7% of the image size) are also applied during training. The proprioceptive state includes the manipulator’s joint angles and the end-effector pose. Specifically, the end-effector’s pose is represented with a 3D translation component and a 6D representation [33] for the rotation. The actions are the end-effector 6 Dof velocities.

Regarding the proposed method, the negative sample queue size, momentum  $m$  and temperature  $\tau$  for the contrastive pipeline are set to 4096, 0.95, and 0.1, respectively, and the preferential weight  $\lambda$  in Eq.(12) is set to 0.05. Both the RcP image encoder and the regression image encoder are initialised from weights pre-trained on ImageNet1K [29]. To enhance the training efficiency of the contrastive pipeline, a warm-up phase is employed. During this phase, slightly shifted versions of the image  $\mathbf{x}$  are used for the initial 5 epochs before transitioning to training with strongly augmented sample pairs as detailed in Sec. III-B.

### B. Simulated Experiments

The simulation environment is created in CoppeliaSim with PyRep [34]. We use a UR5 as the robot manipulator, with up to three Spam Cans as the target instances and up to five non-target random objects (such as bananas, mugs etc). These objects are randomly positioned and re-oriented within the  $0.3 \times 0.5 \text{ m}^2$  workspace. The end-effector’s translation is randomised within a  $0.4 \times 0.2 \times 0.2 \text{ m}^3$  cuboid. Meanwhile, its rotation is adjusted within a downward-pointing cone. Overall, the dataset comprises 3000 trajectories, averaging 20 data points per trajectory. We adopt the multi-instance reaching trajectory generation formulation as described in [17]. Convergence itself is defined as the pose at which the velocity change falls below a predetermined threshold. A reach is deemed successful when the tool-central point is located within a  $3 \text{ cm}^3$  cuboid centred at the target pose, and the deviation in the yaw angle is less than  $10^\circ$ . Notably, within the workspace’s central area for this evaluation, a one-pixel displacement corresponds to a displacement of 1.12 cm.

1) *Robustness against Distractors:* As illustrated in Fig. 4a, we evaluate the emergence of body ownership by introducing various out-of-distribution distractors into the scene: a self-distractor, a Franka Panda robot, and a static sizable object, such as a pot plant. We position each distractor in four distinct locations relative to the robot: behind it towards the left (with partial visual overlap),

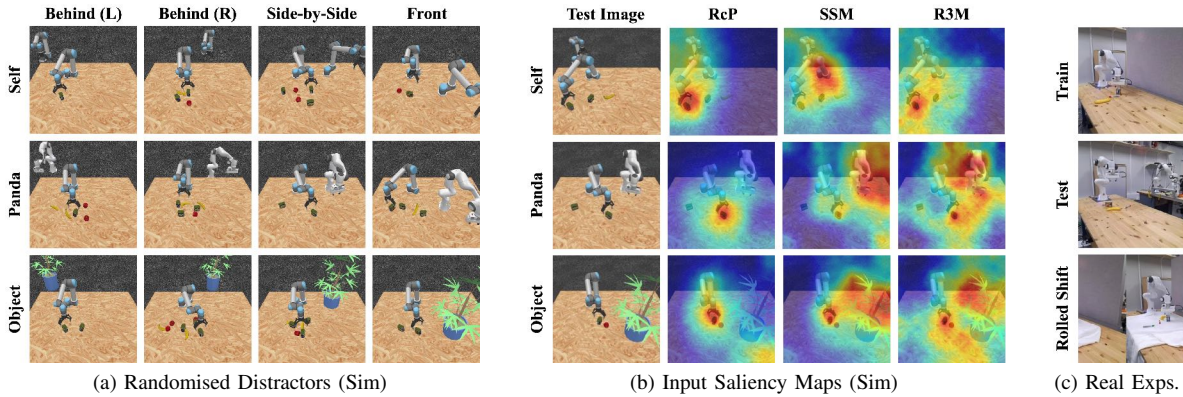


Fig. 4: **Illustration of Simulated Experiments, Input Saliency Maps, and Real Experiments.** (a): Three distractors: the self-distractor, a Franka Panda robot, and a static object (a pot plant), are positioned at four distinct locations: behind the robot towards the left, behind the robot towards the right, alongside the robot, and in front of the robot. During the experiments, both the self-distractor and the Franka Panda execute random actions. (b): We employ an image saliency visualisation tool, FullGrad [6], to visualise the activated image regions for different policies. *RcP*: *Robot-centric Pooling*, *SSM*: *Spatial-Softmax*. (c): The real-world setup features a second-person camera view, with the robot dominant on the left side of the image. A movable divider can conceal and reveals the distractor robot against a less-structured background based on scenarios. The rolled-shift image is used for testing the networks’ robustness against image shifts.

behind it towards the right, alongside the robot, and directly in front of the robot. During tests, both the self-distractor and the Franka undertake random actions. The performance of each network is evaluated by averaging the outcomes across three different random seeds, with each instantiation of the network executing 50 trajectories.

Without distractors, Robot-centric Pooling, Spatial-Softmax and R3M achieve 90.0%, 86.0% and 71.0% reaching success rate, respectively. Under the presence of distractors, as tabulated in Tab. I, Robot-centric Pooling (RcP) demonstrated superior performance across various settings, outperforming Spatial-Softmax (SSM) and R3M in its ability to handle different types of unseen distractors at various locations. Specifically, RcP achieved an average success rate of 76.2% against self-distractors, 70.3% when faced with the Franka Panda robot, and 82.7% against a static plant distractor. On the other hand, the SSM baseline showed variability, with its highest success rate being 62.7% against the Franka Panda when positioned at the back left but fails entirely when the distractors are in the front position.

We also employ an image saliency visualisation tool, FullGrad [6], for analysing the varying performance among different pooling methods. The saliency maps in Fig. 4b from the Robot-centric Pooling (RcP) model exhibit focused attention on the robot and its target, emphasising RcP’s effectiveness in isolating and utilising relevant features for regression tasks. In contrast, saliency maps from Spatial-Softmax and R3M models show less discrimination, erroneously blending features from both the robot and nearby distractors into regression.

As shown in Fig. 4b, when the self-distractor overlaps with the robot, the RcP network may take parts of the distractor for policy regression, hence the relatively lower success rate compared to other distractor locations. This issue is

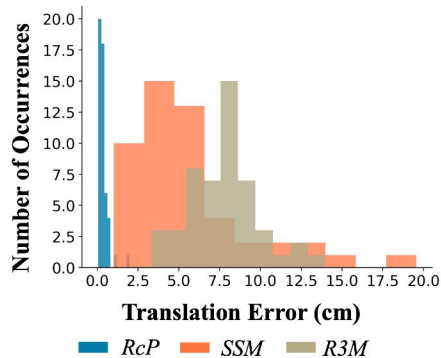
TABLE I: **Reaching Success Rate (%) against different types of distractors at varying locations.** *RcP*: *Robot-centric Pooling*, *SSM*: *Spatial-Softmax*.

		Back(L)	Back(R)	Side	Front	Average
Self	RcP	62.7 ± 8.2	86.7 ± 1.7	81.3 ± 2.5	74.0 ± 0.8	76.2 ± 12.6
	SSM	57.3 ± 6.5	50.7 ± 7.6	2.7 ± 1.9	0.0 ± 0.0	27.7 ± 28.4
	R3M	7.3 ± 2.1	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	1.8 ± 3.8
Franka	RcP	56.0 ± 10.2	78.0 ± 2.2	80.0 ± 4.5	67.3 ± 5.2	70.3 ± 15.7
	SSM	62.7 ± 7.4	17.3 ± 10.9	0.0 ± 0.0	0.0 ± 0.0	20.0 ± 28.8
	R3M	12.0 ± 5.1	8.0 ± 4.3	0.0 ± 0.0	0.0 ± 0.0	5.0 ± 8.5
Plant	RcP	82.0 ± 2.9	85.3 ± 2.1	82.0 ± 2.9	81.3 ± 0.9	82.7 ± 5.0
	SSM	78.0 ± 1.4	84.7 ± 2.1	29.3 ± 8.7	0.0 ± 0.0	48.0 ± 36.1
	R3M	49.3 ± 4.0	6.7 ± 1.2	16.7 ± 6.2	0.0 ± 0.0	18.2 ± 20.4
Total	RcP	66.9 ± 9.5	83.3 ± 2.7	81.1 ± 3.5	74.2 ± 4.2	76.4 ± 13.0
	SSM	66.0 ± 7.2	50.9 ± 15.8	10.7 ± 8.4	0.0 ± 0.0	31.9 ± 33.5
	R3M	22.9 ± 10.2	4.9 ± 3.1	5.6 ± 5.3	0.0 ± 0.0	8.3 ± 14.7

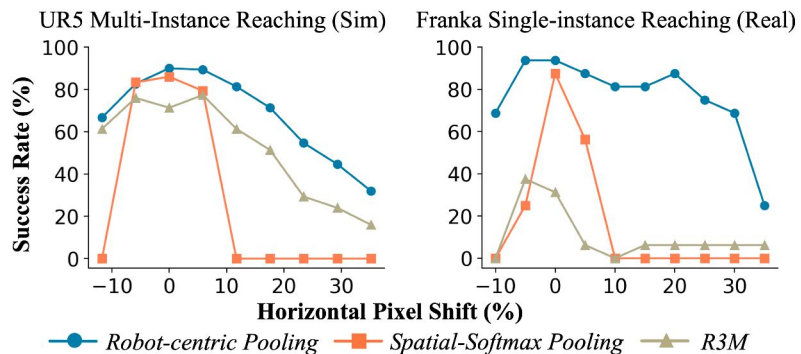
hypothesised to stem from the inherent resolution constraints at the final feature map level, where a  $224 \times 224$  input image is reduced to a  $7 \times 7$  feature map by ResNet18, potentially insufficient for disentangling overlapping features with high precision. We leave further investigation for future work.

### C. Real-World Experiments

As shown in Fig. 4c, the real-world setting features a side second-person camera view, with the robot dominant on the left side of the image. A movable divider conceals and reveals the self-distractor robot against a less-structured background. During the reaching tasks, the robot starts from a predefined home position, targeting a Spam Can that is randomly positioned and reoriented within a designated area of  $0.3 \times 0.3\text{m}^2$ . We collect 200 real-world trajectories combined with 1000 simulated trajectories for training. The networks are first trained with the mixed datasets and then fine-tuned based on real-world data. The training protocol



(a) Translation Error After Self-Distractor's Presence



(b) Reaching success vs percentage of pixel shift.

Fig. 5: **(a): The Translation Error after Self-Distractor's Presence (50 Target Poses).** The translation error is measured at the robot's tool-central point upon convergence, comparing trajectories aimed at the same target with and without a self-distractor. **(b): Reaching success vs percentage of pixel shift.** UR5 multi-instance reaching experiment (Left) and Franka single-instance reaching task (Right).

is identical to the procedure described in Sec. IV-A. Each network is selected out of three random seeds, based on their reaching performance (without distractors) in the simulator.

For evaluating the reaching success and the effectiveness of RcP's robustness against distractions, we position the target at 50 random locations. For each target location, we collect 6 reaching trajectories in total, one reaching attempt from each network (RcP, SSM, and R3M), with and without revealing the self-distractor. A trajectory is deemed successful if the robot can establish a firm grasp on the target by extending 3 cm downward and then closing the gripper.

Without the presence of the self-distractor, the reaching success rate is 96.0%, 90.0%, and 40.0% for RcP, SSM and R3M, respectively. When introducing the self-distractor, RcP demonstrates remarkable robustness, with only a 4% decrease in its success rate. On the other hand, both SSM and R3M exhibit dramatic declines in performance, plummeting to 14% and 2%, respectively.

The histogram presented in Fig. 5a outlines the deviation in translation at the robot's tool-central point upon convergence, comparing trajectories aimed at the same target with and without the presence of a self-distractor. For Robot-centric Pooling (RcP), deviations largely stay within a compact 1cm range, highlighting RcP's consistency. On the other hand, Spatial-Softmax and R3M baselines exhibit significantly higher mean errors and wider error distributions, underscoring the pronounced negative impact of the distractor on their performance.

#### D. Robustness against Pixel Shift

Through both simulated UR5 multi-instance reaching tasks and real-world Franka Panda single-instance reaching tasks, the proposed Robot-centric Pooling (RcP) method demonstrates consistently superior resilience against aggressive horizontal pixel shifts compared to the baseline method. In both the simulation and real-world experiments, the input images are subject to rolled shifts ranging between -10% and +35%, with increments of 5%. The rolled-shift operation

wraps the pixels extending beyond one edge of the image back onto the opposite edge (Fig. 4c). In the simulation, for each shift increment, the performance of each network is averaged across three different random seeds, with each instantiation of the network executing 50 trajectories. The target's pose shuffled for each trajectory. In the real-world experimental setup, the target's pose undergoes 16 random shuffles. For each of these poses, every network carries out one reaching attempt for each shift increment.

As shown in Fig. 5b, RcP maintains a gradually declining success rate in both simulated and real-world experiments, yet significantly outperforms baseline lines. We attribute this inherent robustness against pixel shift to RcP's self-referential nature, which directs the focus of perception from a broad, global view to a localised, robot-centric perspective. In the simulation, R3M exhibits a lower yet similar performance curve as RcP. This resemblance is likely due to the rolled image shift's characteristic of maintaining the overall pixel intensity distribution, which aligns well with the nature of R3M's global average pooling method. However, R3M exhibits high sensitivity to the sim-to-real domain gap. Spatial-Softmax demonstrates a remarkably narrow peak in its performance curve, matching the 7% random pixel shift parameter set during training. This observation suggests its limited capacity for generalisation across varying degrees of pixel shift, indicating the development of strong spatial biases.

#### V. CONCLUSION

In this work, we explored the concept of body ownership within the context of end-to-end visuomotor policy learning. We showed that the conventional end-to-end learning models do not spontaneously develop a sense of body ownership and are highly sensitive to distractors. We introduced Robot-centric Pooling (RcP) that aggregates image features based on image-proprioception alignment. We demonstrated that replacing the conventional final pooling layer with RcP allows the learned policy to develop pronounced self-

recognition and self-other distinction capabilities. Notably, tested with reaching tasks, in both simulated and real-world settings, the policy equipped with RcP exhibited its strong robustness against environmental distractions and self-distractors, significantly surpassing the conventional baselines. Furthermore, benefiting from the robot-centric nature of RcP, the learned policy exhibited enhanced resilience to aggressive image shifts.

In the proposed Robot-centric Pooling, we primarily focused on exploring the spatial alignment aspect for gaining body ownership capability. The potential performance gains by including observational history and robot dynamics have not yet been explored. It entails not only spatial but also temporal alignment, i.e., aligning both ‘seen’ and ‘felt’ positions, velocities, and accelerations, introducing the addition of visuomotor alignment. We leave this exciting direction for future work.

## VI. ACKNOWLEDGEMENT

This work has been supported through WASP - Wallenberg AI, Autonomous systems and Software Program.

## REFERENCES

- [1] K. Kilteni, A. Maselli, K. P. Kording, and M. Slater, “Over my fake body: Body ownership illusions for studying the multisensory basis of own-body perception,” *Frontiers in Human Neuroscience*, 2015.
- [2] M. Synofzik, G. Vosgerau, and M. Voss, “The experience of agency: an interplay between prediction and postdiction,” *Frontiers in psychology*, vol. 4, p. 127, 2013.
- [3] P. Lanillos, G. Cheng, *et al.*, “Robot self/other distinction: active inference meets neural networks learning in a mirror,” in *ECAI 2020*. IOS Press, 2020, pp. 2410–2416.
- [4] B. Yang, D. Jayaraman, G. Berseth, A. Efros, and S. Levine, “Morphology-agnostic visual robotic control,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 766–773, 2020.
- [5] Y. Toshimitsu, K. Kawaharazuka, A. Miki, K. Okada, and M. Inaba, “Dije: Dense image jacobian estimation for robust robotic self-recognition and visual servoing,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022.
- [6] S. Srinivas and F. Fleuret, “Full-gradient representation for neural network visualization,” *Advances in neural information processing systems*, vol. 32, 2019.
- [7] C. Finn, X. Y. Tan, Y. Duan, T. Darrell, S. Levine, and P. Abbeel, “Deep spatial autoencoders for visuomotor learning,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 512–519.
- [8] A. Edsinger and C. C. Kemp, “What can i control? a framework for robot self-discovery,” in *6th International Conference on Epigenetic Robotics*, 2006, pp. 1–8.
- [9] P. Michel, K. Gold, and B. Scassellati, “Motion-based robotic self-recognition,” in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 3. IEEE, 2004, pp. 2763–2768.
- [10] A. Byravan, F. Leeb, F. Meier, and D. Fox, “Se3-pose-nets: Structured deep dynamics models for visuomotor control,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 3339–3346.
- [11] V. Florence, J. J. Corso, and B. Griffin, “Robot-supervised learning for object segmentation,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 1343–1349.
- [12] C. Sancaktar, M. A. van Gerven, and P. Lanillos, “End-to-end pixel-based deep active inference for body perception and action,” in *2020 Joint IEEE 10th International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*. IEEE, 2020, pp. 1–8.
- [13] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín, “What matters in learning from offline human demonstrations for robot manipulation,” *arXiv preprint arXiv:2108.03298*, 2021.
- [14] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, “Diffusion policy: Visuomotor policy learning via action diffusion,” *arXiv preprint arXiv:2303.04137*, 2023.
- [15] T. Xiao, I. Radosavovic, T. Darrell, and J. Malik, “Masked visual pre-training for motor control,” *arXiv preprint arXiv:2203.06173*, 2022.
- [16] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, *et al.*, “Rt-1: Robotics transformer for real-world control at scale,” *arXiv preprint arXiv:2212.06817*, 2022.
- [17] Z. Zhuang, X. Yu, and R. Mahony, “Lyrn (lyapunov reaching network): A real-time closed loop approach from monocular vision,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 8331–8337.
- [18] A. v. d. Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” *arXiv preprint arXiv:1807.03748*, 2018.
- [19] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.
- [20] M. Laskin, A. Srinivas, and P. Abbeel, “Curl: Contrastive unsupervised representations for reinforcement learning,” in *International Conference on Machine Learning*. PMLR, 2020, pp. 5639–5650.
- [21] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, “Momentum contrast for unsupervised visual representation learning,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9729–9738.
- [22] Z. Shen, Z. Liu, Z. Liu, M. Savvides, T. Darrell, and E. Xing, “Unmix: Rethinking image mixtures for unsupervised visual representation learning,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 2, 2022, pp. 2216–2224.
- [23] V. Verma, T. Luong, K. Kawaguchi, H. Pham, and Q. Le, “Towards domain-agnostic contrastive learning,” in *International Conference on Machine Learning*. PMLR, 2021, pp. 10530–10541.
- [24] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, “Unsupervised feature learning via non-parametric instance discrimination,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3733–3742.
- [25] A. Vaswani, “Attention is all you need,” *Advances in Neural Information Processing Systems*, 2017.
- [26] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [27] J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer normalization,” *arXiv preprint arXiv:1607.06450*, 2016.
- [28] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, “Nerf: Representing scenes as neural radiance fields for view synthesis,” *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.
- [29] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE conference on computer vision and pattern recognition*. IEEE, 2009, pp. 248–255.
- [30] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta, “R3m: A universal visual representation for robot manipulation,” *arXiv preprint arXiv:2203.12601*, 2022.
- [31] K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. Liu, *et al.*, “Ego4d: Around the world in 3,000 hours of egocentric video,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18995–19012.
- [32] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” *arXiv preprint arXiv:1711.05101*, 2017.
- [33] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, “On the continuity of rotation representations in neural networks,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5745–5753.
- [34] S. James, M. Freese, and A. J. Davison, “Pyrep: Bringing v-rep to deep robot learning,” *arXiv preprint arXiv:1906.11176*, 2019.