When Pre-trained Visual Representations Fall Short: Limitations in Visuo-motor Robot Learning

Nikolaos Tsagkas[1], **Andreas Sochopoulos**[1] **Duolikun Danier**[1],
Sethu Vijayakumar[1], **Chris Xiaoxuan Lu**[2], **Oisin Mac Aodha**[1]
[1]University of Edinburgh, [2]UCL

Abstract: The integration of pre-trained visual representations (PVRs) into visuo-motor robot learning has emerged as a promising alternative to training visual encoders from scratch. However, PVRs face critical challenges in the context of policy learning, including temporal entanglement and an inability to generalise even in the presence of minor scene perturbations. These limitations hinder performance in tasks requiring temporal awareness and robustness to scene changes. This work identifies these shortcomings and proposes solutions to address them. First, we augment PVR features with temporal perception and a sense of task completion, effectively disentangling them in time. Second, we introduce a module that learns to selectively attend to task-relevant local features, enhancing robustness when evaluated on out-of-distribution scenes. We demonstrate significant performance improvements, particularly in PVRs trained with masking objectives, and validate the effectiveness of our enhancements in addressing PVR-specific limitations. Project page: `tsagkas.github.io/pvrobo`

Keywords: Pre-trained Visual Representations, Behaviour Cloning

1 Introduction

Performing robust and accurate robotic manipulation from visual inputs necessitates informative and stable visual representations. The traditional paradigm for training visuo-motor policies has involved learning visual encoders from scratch alongside policy models [1]. Recently, however, the adoption of pre-trained visual representations (PVRs), *i.e.,* computer vision models trained on large and diverse visual datasets, has emerged as a compelling alternative, moving away from the tabula-rasa approach [2]. This shift has been driven by three key factors: the SoTA performance of PVRs in computer vision tasks, their impressive generalisation capabilities derived from training on vast datasets, and the absence of robust robot-specific foundation models capable of addressing challenges unique to robotics, such as handling diverse embodiments.

Despite the promising results of PVRs in downstream robotic applications, including affordance-based manipulation [3], semantically precise tasks [4], and language-guided approaches [5], their effective integration into visuo-motor policy learning for even basic pick-and-place tasks remains an open challenge. Crucially, training visual encoders from-scratch or fine-tuning them with in-domain data still leads to competitive performance compared to using raw PVR features or even adapted PVRs [6, 7]. Furthermore, no single PVR, or set of characteristics, has consistently delivered optimal performance across diverse tasks and environments [8, 9]. Notably, their generalisation capabilities remain underutilised, as small scene variations can destabilise policy models [10, 11], see Fig. A7. This limitation has reignited interest in training models from scratch with augmented datasets [7], a strategy that is prohibitively expensive for many real-world applications.

We identify critical shortcomings in the current use of PVRs for visuo-motor policy learning, rooted in the inherent nature of PVR features. First, we observe that these features are temporally entangled, primarily because widely used PVRs are designed as time-invariant models. Additionally, imitation learning datasets often consist of frame sequences where only minor changes occur in the pixel
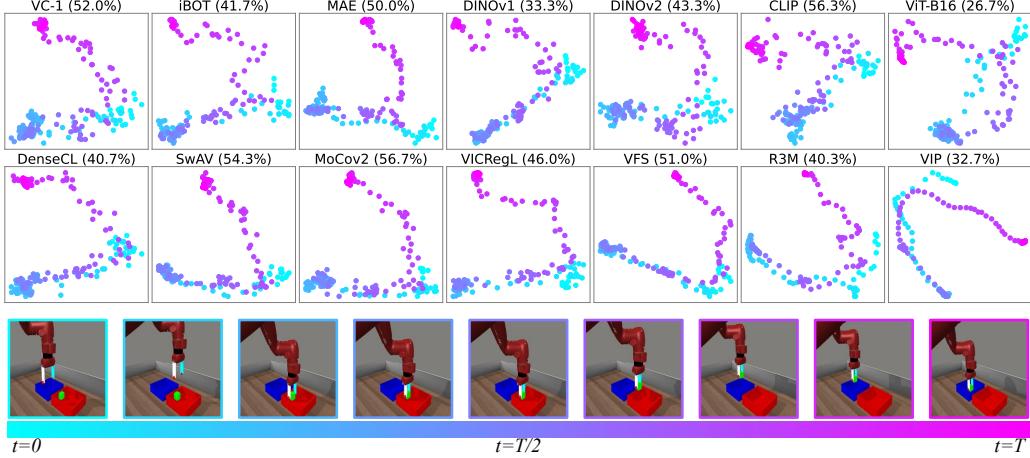
Figure 1: PCA of features from an expert demonstration in Bin Picking across PVRs (Top row: ViT models; Middle row: ResNet models). Frame colours align with trajectory stages, suggesting feature entanglement during the gripper **descent** and **ascent**, and during the **gripper stop** phase. Our disentangling method (Sec. 3.2) improves success by 16.4% on the Bin Picking task, up from <50%.

domain between adjacent timesteps. As a result, the extracted features from these frames remain highly similar, even at transition points where the corresponding actions may differ significantly. This discrepancy forces policy models to map nearly identical inputs to divergent outputs, introducing a problematic one-to-many mapping that violates the Markov property (see Fig. 1). Second, policy networks tend to overfit to features corresponding to dominant but irrelevant static visual cues (*e.g.,* background elements), making them overly sensitive to minor scene perturbations. These seemingly small prediction errors accumulate over time, leading to substantial performance degradation.

The aforementioned PVR characteristics are key factors hindering visuo-motor policy learning and we argue that these issues should be addressed at the feature level. Attempting to resolve the temporal entanglement problem within the policy network would limit the flexibility of PVRs in general policy architectures. For example, an LSTM policy network could be a good candidate for that [12], but would prevent us from conditioning other SoTA approaches, *e.g.,* diffusion policies [13]. Similarly, augmenting the dataset for improving the policy's robustness [7] would be prohibitive for real-world robot applications, as it would require a great number of man-hours and the fine-tuning of PVR weights could affect the rich encoded knowledge.

In summary, we make the following contributions:

1. Identifying PVR limitations. We identify key characteristics of PVRs that hinder effective visuo-motor robot learning. Specifically, we show that they fail to encode the temporal cues and scene agnostic fine-scale visual features needed for precise manipulation.

2. Temporal disentanglement. We enhance PVR features by incorporating temporal awareness and task-completion perception, without altering the policy model architecture, yielding a statistically significant improvement in downstream task performance. We also show that our method improves performance even in models that already incorporate temporal structure, whether in the input (via context with rotary positional embeddings) or the output (via action chunking [14]).

3. Targeted visual features. We introduce a module that learns to directly attend to task-relevant visual cues while ignoring scene distractors. Our approach does not require dataset augmentation or re-training but instead more effectively utilises existing PVR features, particularly benefiting masked image modelling (MIM) trained PVRs. Our approach consistently outperforms existing widely-used token pooling strategies (*e.g.,* Spatial Softmax [15] or TokenLearner [16, 17]), under different scene variations and perturbations.

2 Related Work

PVRs in Visuo-motor Policy Learning. In [2], frozen PVRs were evaluated across simulated environments, outperforming models trained from scratch. Similarly, [9] showed that the utility of PVRs depends on the policy training paradigm, with behaviour cloning and inverse reinforcement learning yielding robust results, while reinforcement learning exhibited higher variability. Furthermore, [18] provided evidence that simulation experiments (*e.g.,* MetaWorld [19]) are indicative of real world performance for PVR-based trained policies.

PVRs are favoured for their generalisation capabilities in vision tasks, but out-of-distribution generalisation remains challenging in policy deployment. [11] analysed the impact of various perturbations on PVR-based policy generalisation, while [20] identified correlations between generalisation performance and inherent model traits, such as ViTs' segmentation ability. Conversely, [7] found that learning from scratch with data augmentation can yield competitive results, while [21] found that adapters [22] can improve policy generalisation when training with diverse object instances. We focus on developing methods that achieve robustness to scene changes without relying on dataset augmentation, which can be prohibitively expensive in real-world robotics applications.

Time-informed Policy Training. In PVR-based visuo-motor policy learning, the incorporation of temporal information remains underexplored. Augmentation with temporal perception can happen either at feature level or during training time. While early fusion methods, such as stacking multiple frames before encoding [23], are common in training visual encoders from scratch, late fusion (*i.e.,* processing frames individually and stacking their representations [24]) has shown superior performance with fewer encoder parameters. Recent work [25] highlights that naive feature concatenation in latent space is insufficient; instead, approaches like FLARE [25] incorporate sequential embeddings and their differences, inspired by optical flow techniques. Nevertheless, concatenating sequential embeddings as input to policy networks has become standard in visuo-motor policy learning [2] and SoTA generative policies [13, 26]. However, a gap remains in leveraging PVR features, which are primarily designed for vision tasks, within this temporal framework.

A major limitation of many PVRs is their inherit lack of temporal perception, as most are pre-trained on static 2D image datasets. Temporal perception can be added by employing loss functions that enforce temporal consistency during training (*e.g.,* R3M [27] and VIP [28]), when training with video data. However, there is no clear consensus on the superiority of this approach compared to alternatives like MIM (*e.g.,* MVP [29, 9] and VC-1 [8]). This disparity suggests that existing temporal modelling strategies may be insufficient in isolation. In later experiments, we evaluate PVRs trained with temporal information and demonstrate that methods trained with a time-agnostic paradigm achieve comparable performance. We hypothesise that this limitation arises from a lack of task-completion perception, which we address by incorporating positional encoding—a fundamental mechanism in many machine learning approaches. This straightforward operation has been instrumental in the success of Transformers [30], implicit spatial representations [31], and diffusion processes [32].

Task-Relevant Feature Extraction. Downstream vision tasks often make use of the output features of PVRs. However, these features typically encode a broad range of scene information, much of which may be irrelevant to the specific task. To address this challenge, attentive probing [33, 34, 35] has emerged as a popular evaluation technique, leveraging local tokens. This approach leverages a cross-attention layer with a trainable query token, treating the local features from PVRs as a sequence of key-value pairs. Unlike traditional evaluation methods such as linear probing, attentive probing has shown significantly different vision evaluation outcomes, particularly with PVRs trained using MIM approaches (*e.g.,* MAE [36]), where features, such as the CLS token, often include irrelevant information.

In robotics applications, pooling a sequence of observation tokens is not a novel idea and it is usually deployed for reducing the input stream length. RT-1 [17] utilised TokenLearner [16] for making the policy input more compact, thus speeding up inference time. Concurrently to our work, ICRT [37] deployed an attentive pooling module for the purpose of extracting a single state token that summarizes proprioception with visual observations. However, there is currently no consensus
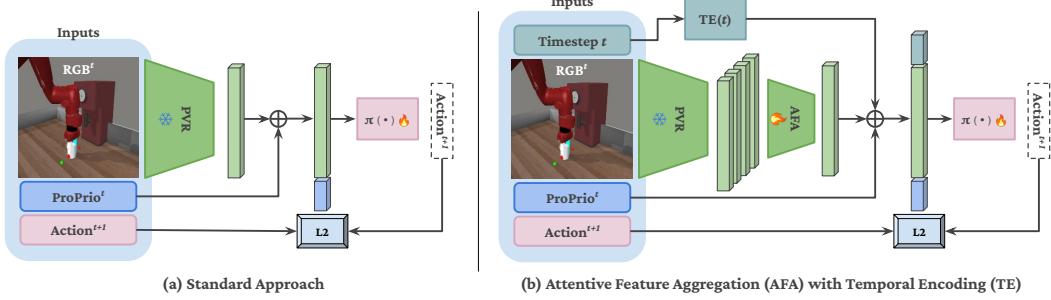
Figure 2: Standard PVR-based visuo-motor policy learning via behaviour cloning (a) and our approach (b), which integrates *Temporal Encoding (TE)* for temporal features (Section 3.2) and *Attentive Feature Aggregation (AFA)* for selective local feature attention (Section 3.3).

as to how PVR features can be effectively used such that the policy model learns to attend only to task-relevant signals, ignoring unrelated cues that act as distractors. By prioritizing task-relevant signals, we show that attentive probing enhances task performance, especially in out-of-distribution scenarios.

3 Methodology

3.1 Preliminaries: Imitation Learning via Behaviour Cloning

We consider an expert policy $\pi^\star : \mathcal{P} \times \mathcal{O} \to \mathcal{A}$, which maps a robot's proprioceptive observation $p \in \mathcal{P}$ and visual observation $o \in \mathcal{O}$ to an action $a \in \mathcal{A}$. This policy generates a dataset $\mathcal{T}^e = \{(p_t^i, o_t^i, a_t^i)_{t=0}^T\}_{i=1}^N$ of N expert trajectories, where each trajectory contains T steps of observations and actions for a task.

We employ behaviour cloning to learn a policy π_θ, parameterised by θ, to imitate $\pi^\star$ by minimizing the action discrepancy over demonstrations: $\mathbb{E}_{(p_t^i, o_t^i, a_t^i) \sim \mathcal{T}^e} \|a_t^i - \pi_\theta(f_{\text{PVR}}(o_t^i), p_t^i)\|_2^2$, where f_{PVR} is a pre-trained visual representation (PVR) that extracts features from o_t^i. In visuo-motor policy learning, it is common to assume the Markov property, whereby the current observation $x_t = (p_t, o_t)$ suffices for predicting the next state: $P(x_{t+1}|x_t) = P(x_{t+1}|x_t, x_{t-1}, \ldots, x_0)$. This allows tasks to be modelled as Markov decision processes, where each action depends only on the current state, enabling the use of behaviour cloning under this formulation.

3.2 Temporal Disentanglement

We observe that the assumption of Markovian decision-making in policies using features from frozen PVRs is often invalid. This arises because, at each timestep, the available information may be insufficient for the policy to confidently map the current observation to the appropriate action.

Consider the example presented in Fig. 1, where PVR-features of the same pick-and-place trajectory are projected with PCA into 2D. Regardless of the PVR utilised, the extracted features seem to suffer from temporal entanglement. First, features extracted from the frames where the robot has stopped to pick up the box often form a tight cluster, since the only change is the movement of the gripper fingers, which corresponds to a very small percentage of pixels. Second, as the gripper moves down and subsequently ascends, the primary visual change is the cube's vertical displacement relative to the table. Consequently, the visual features extracted from the descent and ascent frames may differ only marginally, and only in dimensions affected by the small pixel region of the cube.

Training a policy network to map (p_t, o_t) to a_t becomes difficult under these conditions. When multiple observations are nearly indistinguishable, the mapping violates the functional requirement that each input must map to exactly one output. To address these challenges, we propose a simple yet effective method to augment each observation with a temporal component by encoding the timestep index of each frame as a high-dimensional vector, using $\gamma(t) =$

$\left(\sin\left(\frac{2^0 \pi t}{s^0}\right), \cos\left(\frac{2^0 \pi t}{s^0}\right), \ldots, \sin\left(\frac{2^{T-1} \pi t}{s^{T-1}}\right), \cos\left(\frac{2^{T-1} \pi t}{s^{T-1}}\right)\right)$ and concatenating to the policy input (see Fig. 2). This augmentation can temporally disentangle similar (p_t, o_t) pairs, introducing a task progression signal into the robot state, which we argue can enhance policy performance.

Thus, we encode each timestep t into a high-dimensional vector $\gamma(t)$ using alternating sine and cosine functions at exponentially increasing frequencies 2^k. The lower-frequency terms capture coarse temporal trends, while the higher-frequency terms provide finer temporal resolution, enabling the policy to distinguish between temporally similar states. Note that traditionally such embeddings encode the relative position in a transformer's input [38], whereas here we encode the position of the embedding in the rollout.

3.3 Attending to Policy Relevant Features

We posit that training policies using the global features of PVRs (*i.e.*, CLS token for ViTs or average pooled channel feature for CNNs) can lead to overfitting to scene conditions that are irrelevant to the task at hand. The output features of these representations often capture visual characteristics of the scene that may be irrelevant to the policy (*e.g.*, the texture of a tabletop). Processing such extraneous information not only dilutes the policy network's focus, but also leads to overfitting to scene specific conditions. This observation aligns with recent work on vision model evaluation [33], which argues that only specific image regions carry the necessary information for solving a task. Building on this insight, we hypothesise that incorporating local information is particularly effective in the context of robot learning, echoing findings in PVR distillation research [39], though this area remains empirically underexplored.

Recognising the importance of local information is only part of the solution. A data-driven mechanism is also required to filter irrelevant details, such as background patches, and prioritise task-relevant information. To this end, we adopt the *attentive probing* methodology [33] to implement *Attentive Feature Aggregation* (AFA). Specifically, we append a cross-attention layer to the frozen PVR, modified to include a trainable query token that interacts with the sequence of local tokens produced by the model. These tokens correspond to the per-patch embeddings for ViTs and the channel embeddings for CNNs, both from the final layer.

Consequently, we deploy: $Attention(q, F) = softmax\left(\frac{q \cdot (F \cdot W_K)^\top}{\sqrt{d_k}}\right) F \cdot W_V$. The query token q computes dot products with the feature sequence, with length equal to the number of patches and dimension d_k, organized as a matrix F. These dot products are passed through a softmax function to assign weights to the contributions of each local token to the final embedding. Our AFA module consists of multiple heads, so that specific dimension groups that might be irrelevant to the policy can be filtered out. Gradients are allowed to flow through the cross-attention layer, updating the parameters of q as well as the key and value projection matrices, W_K and W_V.

4 Experiments

4.1 Implementation Details

Environment. We conduct our experiments in the widely used MetaWorld simulation environment [19], which is built on the MuJoCo [40] physics engine. From this benchmark, we select the ten tasks visualised in Fig. A7 and generate 25 expert demonstrations with a maximum of 175 rollout steps for each using the provided heuristic policies. The primary criterion for task selection is to maintain a balanced representation of easy, medium, and hard tasks, as identified in prior work on PVR-based visuo-motor control [41, 9], as well as from our empirical results.

PVRs. To validate our hypotheses, we deploy seven Residual Networks (ResNets) [42] and seven Vision Transformers (ViT) [43], as summarised in Tab. A1 in the Appendix. Our selection includes the most popular PVRs utilised in robot learning applications that have led to SoTA performance. We also aim to ensure diversity across different training strategies, datasets, and the balance between local and global perception. Despite these variations, we maintain a consistent backbone architecture
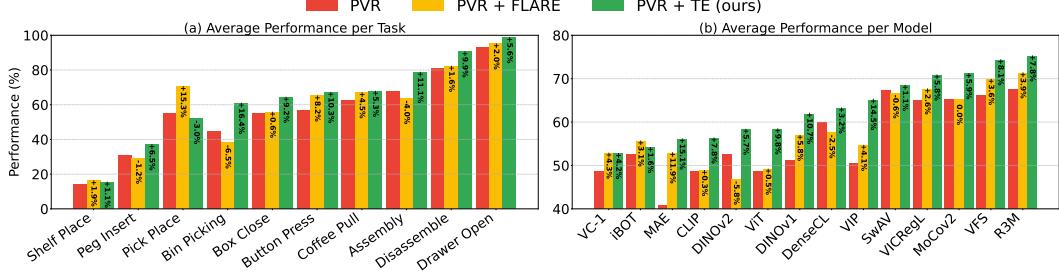
Figure 3: Comparison of our Temporal Encoding (TE) against FLARE [25] and using no temporal augmentation on PVR features. Results (sorted by TE) show (a) per-task performance and (b) per-model performance. FLARE and TE bars indicate gains over no temporal information.

of ResNet-50 or ViT-B/16, with the exception of DINOv2 [44], which employs a smaller patch size of 14. Also, for DINOv2 we discard overlapping patches, to ensure fairness in the comparison of PVRs. The models tested include powerful representations from vision-specific approaches (*e.g.,* DINO [10]), vision-language models (*e.g.,* CLIP [45]), and robot-learning-focused models (*e.g.,* R3M [27]).

Policy training. For all policy training, we repeat the policy network training five times using different seeds, keeping the PVR frozen, and report the interquartile mean (IQM) success rate. As is common practice in similar work [2, 27, 9], our policy head consists of a small number of MLP layers (four in our case), separated by ReLUs, and outputs the predicted action. We train with mini-batches of 128 samples for 80,000 steps.

4.2 Temporal Encoding

We evaluate the performance of a policy network trained for each PVR under three conditions: (i) without any temporal component, (ii) using the three most recent past observations and their latent differences (*i.e.,* FLARE [25]), and (iii) with our proposed Temporal Encoding (TE) method. We select the dimensionality of TE to be 64 and the scale parameter 100, after tuning these hyperparameters in a subset of tasks and PVRs (see Appendix C.3).

Fig. 3 illustrates (a) the average performance per task for each of the three approaches and (b) the average performance per model. Statistical analysis (paired t-tests, Wilcoxon tests-results in Appendix C.2) confirms that our TE's gains over FLARE and no augmentation are significant. While VC-1 and iBOT achieve slightly higher average scores with FLARE, all other PVRs benefit significantly from temporal augmentation, even when compared to FLARE-augmented results. Similarly, apart from the "Pick and Place" and "Shelf Place" tasks, TE significantly enhances the average task performance. We attribute TE's superiority over FLARE to the fact that FLARE's method of stacking sequential latent embeddings, along with their differences, results in embeddings that are nearly identical due to the high similarity of features extracted from consecutive observations. Consequently, the differences between these embeddings are often near-zero vectors, limiting FLARE's ability to effectively leverage temporal information. Table A2 provides a detailed summary of the performance of each model-task pair.

An intriguing observation is that PVRs pre-trained with a temporal component in their objective function also benefit considerably from TE. For instance, R3M employs time-contrastive learning [46] to enforce similarity between representations of temporally adjacent frames-experience substantial gains from TE. In addition, VIP achieves an average performance boost of approximately 15%, making it one of the most positively impacted models. This finding suggests a potential reconsideration of how temporal perception is integrated into features designed for robot learning. It raises the possibility that existing approaches may not fully exploit the temporal structure necessary for optimal performance.

Finally, we also evaluate our TE in conjunction with popular policy modelling techniques that implicitly capture temporal structure. Specifically, we integrate a causal transformer [47, 37] into
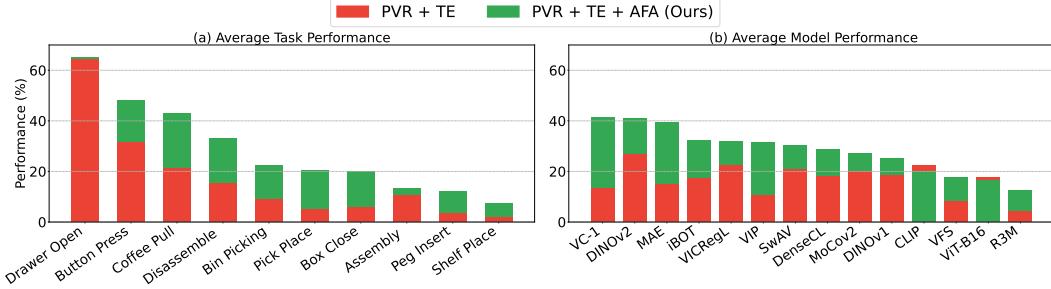
Figure 4: Robustness evaluation of TE-augmented PVR features, with and without our AFA feature extraction, under lighting and texture perturbations. Subplot (a) reports average performance per task; (b) shows per-model averages.

our policy model and extract features with the strong-performing R3M. This transformer processes a sequence of input tokens (context) and produces a sequence of future actions (horizon). We use rotary positional embeddings [38], which encode the relative position of tokens within the input context, unlike our TE, which encodes the position of each token within the rollout. We apply TE at the output stage of the transformer. To emphasize that TE serves to disentangle temporally adjacent tokens, rather than encode absolute timesteps, we train a multitask policy model across four tasks, using context and horizon length equal to 12. Even with a single attention layer and only 25 demonstrations per task, the inclusion of TE led to higher success rates, as shown in Tab. A6.

4.3 Attentive Feature Aggregation

We use a cross-attention layer with 12 heads for ViT features and 32 heads for ResNet features. This configuration ensures that we process 64-dimensional feature chunks in both cases, maintaining fairness between the two backbone architectures. Consistent with standard attentive probing methodologies [33, 34], we employ a cosine learning rate scheduler with a warm-up phase. TE is applied to augment features with a temporal component and specifically for AFA, the temporal feature is concatenated with the output of the cross-attention module. We validate our full model's significance via an ablation study, presented in Fig. A4, that evaluates the contributions of TE and AFA in our approach, comparing it against training the policy network solely with PVR-extracted features.

We hypothesise that AFA learns to attend only to scene areas that are important to the task, disregarding features irrelevant to the policy. To properly evaluate this component, we do not limit our evaluation to in-domain scenes but also with environments recreated with scene perturbations, leaving the training dataset distribution unchanged. These perturbations include two modifications, details of which are included in Appendix D.2. First, tabletop texture changes, randomly selected from 30 distinct textures, some of which feature vibrant patterns that act as strong distractors. These changes affect a significant area of the frame. Second, variations in lighting conditions, including adjustments to the orientation, position, and brightness of the light source. These modifications influence the entire frame, including the robot and the object it manipulates. We summarize the performance of policies trained directly with PVR features and those trained with features aggregated using AFA in Fig. 4, averaged across tasks and models on perturbed scenes. In Fig. A5 we also provide a summary of the performance of each task and model pair on in-domain scenes and on perturbed scenes (per perturbation category).

From these results, several trends emerge. AFA consistently improves the robustness of policies across all tasks, often yielding significant gains, with some tasks showing up to a threefold improvement. The only exception is the "Drawer Open" task, which sees little benefit from AFA. This is likely due to its inherent simplicity, as the task involves manipulating a large object without requiring control of the gripper fingers, which remain open throughout the demonstrations. Consequently, attending to local observations has limited impact. Additionally, most models exhibit improved out-of-distribution performance with AFA, except for CLIP and ViT-B/16. This is reasonable, as these models are trained with objectives that emphasise global frame perception, unlike other models that incorporate

7

supervision at the patch level. Notably, MIM-trained PVRs (*i.e.,* VC-1, DINOv2, MAE and iBOT) benefit the most from AFA, reflecting the alignment between AFA's design, which is inspired by attentive probing, and the training principles of MIM-based models. These findings highlight AFA's ability to enhance policy performance, particularly in challenging out-of-distribution scenarios, and underscore its compatibility with models that leverage local feature representations.

The average in-domain performance remains nearly unchanged, with a slight increase from 63.1% to 66.4% when using AFA. The minor boost observed with AFA in in-domain scenarios, especially when compared to its substantial improvements in perturbed scenes, suggests that AFA does not learn a new latent space for the PVR, more suited to the task. Instead, it appears to refine the use of the existing latent space by learning to leverage relevant information while discarding elements that are irrelevant to the policy. This distinction underscores AFA's role as a mechanism for better utilisation of pre-trained features rather than redefining or adapting the underlying feature space.

To further validate that the observed performance boost stems from the use of AFA, and not due to simply pooling local features, we evaluate against two popular alternatives. First, we combine features with Spatial Softmax [15], where we compute the expected 2D coordinates of feature activations across the spatial grid for each channel, effectively summarising the feature map into a set of soft keypoints. Second, we apply TokenLearner [16] configured to extract a single latent token, where a learned spatial attention map is used to selectively aggregate information across all spatial positions into one content-rich representation. We examine both alternatives on all tasks for the top-performing PVR (*i.e.,* VC-1), and we observe (Fig. A10) that AFA leads to much greater out-of-distribution performance, outperforming both by more than 20%.

5 Discussion

We explored how to effectively utilise PVRs for visuo-motor policy learning by identifying the issues of feature temporal entanglement and lack of robustness in scene visual changes. Furthermore, we proposed two approaches to address these limitations leading to a significant performance increase.

Augmenting PVR-extracted features with TE significantly improved performance, even when combined with SoTA methods that implicitly model temporal structure (*i.e.,* causal transformers with context and action horizons lengths > 1). We argue that this gain is not due to properties of the heuristic expert demonstrations, which are asynchronous, as Fig. A3 shows. Feature entanglement is also unlikely to be simulation-related, since visualizations in Fig. A1 and Fig. A2 replicate Fig. 1 using real-world demos and reveal similar behaviour. We also evaluated popular video-PVRs, expecting improved temporal modelling. However, as Table A5 indicates, their features were less effective than those from image-PVRs, and still benefited from TE. This counter-intuitive result aligns with the findings of [13], regarding the length of the observation horizon.

Learning to attend to task-related visual cues is imperative for increasing policy robustness under scene perturbations. We argue that this success does not arise from the increase in learning capacity that AFA introduces, since training policy a network using raw PVR features and with a deeper policy network to match AFA's parameter count did not match AFA's performance (see Fig. A8). Also, our performance retaliative to TokenLearner and Spatial Softmax further highlights the robustness of AFA under scene perturbations. Additional validation through real-world robot experiments (Section D.6) further supports the generality of our method.

6 Conclusion

The use of PVRs for visuo-motor policy learning is still in its early stages, and we believe our work highlights key challenges and paves the way for further exploration. In particular, we highlighted characteristics that PVR features currently lack for effective policy learning, such as an inherent sense of task completion and the ability to focus on task-relevant cues. Our insights contribute to the development of PVR models specifically designed for robot learning, ultimately leading to a generalist robotic system powered by large-scale vision foundation models.

Limitations

We conducted extensive experiments, evaluating multiple PVRs and training policies across various seeds and representative tasks. Nonetheless, our study has some limitations.

First, due to the combinatorial scale of the experimental setup—10 tasks, 5 seeds, 14 PVRs, and 4 architectural variants (*PVR-only*, *PVR+TE*, *PVR+AFA*, *PVR+TE+AFA*), which accumulates to 2,800 training sessions, not including the addition experiments reported in the appendix (*e.g.,* causal transformer, other pooling methods, etc.)—our experiments required significant computational resources (multiple GPU days), limiting our ability to explore a broader range of tasks or other simulation environments.

To mitigate this, we adopted two strategies:

1. We selected challenging MetaWorld tasks that were not trivially solvable. For example, we observed empirically that tasks such as `hammer-v2` achieved near-saturated success rates ($\sim$95%) across most PVRs, and thus we omitted it to focus attention on more difficult alternatives.

2. We conducted robustness experiments, similar to the ones in Fig. 4, on a real-world robotic platform to evaluate the transferability of AFA (Section D.6).

Second, while our finding that PVR features exhibit temporal entanglement is significant, its implications in more complex scenarios remain open. In particular, our temporal encoding (TE) approach may struggle to capture meaningful temporal structure in long-horizon tasks involving multiple subgoals or phases. Potentially this problem could be solved by resetting the timestep after each subtask is completed, using a high-level planner to keep track of task progression.

Finally, incorporating local information through AFA introduces a computational trade-off. Since it requires processing sequences of tokens, both training and inference become more resource-intensive, posing challenges for real-time or resource-constrained deployments. Nevertheless, we had no problem running AFA at 10Hz on a NVIDIA GeForce RTX 2080, deploying a trained policy on a KUKA IIWA 14 robot, as discussed in Section D.6.

References

[1] S. Levine, C. Finn, T. Darrell, and P. Abbeel. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research (JMLR)*, 2016.

[2] S. Parisi, A. Rajeswaran, S. Purushwalkam, and A. Gupta. The (un)surprising effectiveness of pre-trained vision models for control. In *International Conference on Machine Learning (ICML)*, 2022.

[3] G. Li, N. Tsagkas, J. Song, R. Mon-Williams, S. Vijayakumar, K. Shao, and L. Sevilla-Lara. Learning precise affordances from egocentric videos for robotic manipulation. *arxiv preprint arXiv:2408.10123*, 2024.

[4] N. Tsagkas, J. Rome, S. Ramamoorthy, O. Mac Aodha, and C. X. Lu. Click to grasp: Zero-shot precise manipulation via visual diffusion descriptors. In *International Conference on Intelligent Robots and Systems (IROS)*, 2024.

[5] W. Shen, G. Yang, A. Yu, J. Wong, L. P. Kaelbling, and P. Isola. Distilled feature fields enable few-shot language-guided manipulation. In *Conference on Robot Learning (CoRL)*, 2023.

[6] M. Sharma, C. Fantacci, Y. Zhou, S. Koppula, N. Heess, J. Scholz, and Y. Aytar. Lossless adaptation of pretrained vision models for robotic manipulation. In *International Conference on Learning Representations (ICLR)*, 2023.

[7] N. Hansen, Z. Yuan, Y. Ze, T. Mu, A. Rajeswaran, H. Su, H. Xu, and X. Wang. On pre-training for visuo-motor control: Revisiting a learning-from-scratch baseline. In *International Conference on Machine Learning (ICML)*, 2023.

[8] A. Majumdar, K. Yadav, S. Arnaud, J. Ma, C. Chen, S. Silwal, A. Jain, V.-P. Berges, T. Wu, J. Vakil, P. Abbeel, J. Malik, D. Batra, Y. Lin, O. Maksymets, A. Rajeswaran, and F. Meier. Where are we in the search for an artificial visual cortex for embodied intelligence? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[9] Y. Hu, R. Wang, L. E. Li, and Y. Gao. For pre-trained vision models in motor control, not all policy learning methods are created equal. In *International Conference on Machine Learning (ICML)*, 2023.

[10] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. Emerging properties in self-supervised vision transformers. In *International Conference on Computer Vision (ICCV)*, 2021.

[11] A. Xie, L. Lee, T. Xiao, and C. Finn. Decomposing the generalization gap in imitation learning for visual robotic manipulation. In *International Conference on Robotics and Automation (ICRA)*, 2024.

[12] M. J. Hausknecht and P. Stone. Deep recurrent q-learning for partially observable mdps. In *Association for the Advancement of Artificial Intelligence (AAAI) Fall Symposium*, 2015.

[13] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems (RSS)*, 2023.

[14] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning fine-grained bimanual manipulation with low-cost hardware, 2023.

[15] C. Finn, X. Y. Tan, Y. Duan, T. Darrell, S. Levine, and P. Abbeel. Deep spatial autoencoders for visuomotor learning. In *International Conference on Robotics and Automation (ICRA)*, 2016.

[16] M. S. Ryoo, A. Piergiovanni, A. Arnab, M. Dehghani, and A. Angelova. Tokenlearner: Adaptive space-time tokenization for videos. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[17] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, T. Jackson, S. Jesmonth, N. J. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, K.-H. Lee, S. Levine, Y. Lu, U. Malla, D. Manjunath, I. Mordatch, O. Nachum, C. Parada, J. Peralta, E. Perez, K. Pertsch, J. Quiambao, K. Rao, M. Ryoo, G. Salazar, P. Sanketi, K. Sayed, J. Singh, S. Sontakke, A. Stone, C. Tan, H. Tran, V. Vanhoucke, S. Vega, Q. Vuong, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich. Rt-1: Robotics transformer for real-world control at scale. In *Robotics: Science and Systems (RSS)*, 2023.

[18] S. Silwal, K. Yadav, T. Wu, J. Vakil, A. Majumdar, S. Arnaud, C. Chen, V.-P. Berges, D. Batra, A. Rajeswaran, M. Kalakrishnan, F. Meier, and O. Maksymets. What do we learn from a large-scale study of pre-trained visual representations in sim and real environments? In *International Conference on Robotics and Automation (ICRA)*, 2024.

[19] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning (CoRL)*, 2020.

[20] K. Burns, Z. Witzel, J. I. Hamid, T. Yu, C. Finn, and K. Hausman. What makes pre-trained visual representations successful for robust manipulation? In *Conference on Robot Learning (CoRL)*, 2024.

[21] X. Lin, J. So, S. Mahalingam, F. Liu, and P. Abbeel. Spawnnet: Learning generalizable visuomotor skills from pre-trained network. In *International Conference on Robotics and Automation (ICRA)*, 2024.

[22] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning (ICML)*, 2019.

[23] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[24] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

[25] W. Shang, X. Wang, A. Srinivas, A. Rajeswaran, Y. Gao, P. Abbeel, and M. Laskin. Reinforcement learning with latent flow. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[26] A. Sochopoulos, N. Malkin, N. Tsagkas, J. Moura, M. Gienger, and S. Vijayakumar. Fast flow-based visuomotor policies via conditional optimal transport couplings. *arXiv preprint arXiv:2505.01179*, 2025.

[27] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta. R3m: A universal visual representation for robot manipulation. In *Conference on Robot Learning (CoRL)*, 2022.

[28] Y. J. Ma, S. Sodhani, D. Jayaraman, O. Bastani, V. Kumar, and A. Zhang. VIP: Towards universal visual reward and representation via value-implicit pre-training. In *International Conference on Learning Representations (ICLR)*, 2023.

[29] T. Xiao, I. Radosavovic, T. Darrell, and J. Malik. Masked visual pre-training for motor control. *arXiv preprint arXiv:2203.06173*, 2022.

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[31] 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. In *European Conference on Computer Vision (ECCV)*, 2020.

[32] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[33] X. Chen, M. Ding, X. Wang, Y. Xin, S. Mo, Y. Wang, S. Han, P. Luo, G. Zeng, and J. Wang. Context autoencoder for self-supervised representation learning. *International Journal of Computer Vision (IJCV)*, 2024.

[34] D. Danier, M. Aygün, C. Li, H. Bilen, and O. Mac Aodha. DepthCues: Evaluating Monocular Depth Perception in Large Vision Models. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[35] A. Bardes, Q. Garrido, J. Ponce, M. Rabbat, Y. LeCun, M. Assran, and N. Ballas. Revisiting feature prediction for learning visual representations from video. *arXiv:2404.08471*, 2024.

[36] K. He, X. Chen, S. Xie, Y. Li, P. Doll'ar, and R. B. Girshick. Masked autoencoders are scalable vision learners. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[37] L. Fu, H. Huang, G. Datta, L. Y. Chen, W. C.-H. Panitch, F. Liu, H. Li, and K. Goldberg. In-context imitation learning via next-token prediction. In *International Conference on Robotics and Automation (ICRA)*, 2025.

[38] J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo, and Y. Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 2024.

[39] J. Shang, K. Schmeckpeper, B. B. May, M. V. Minniti, T. Kelestemur, D. Watkins, and L. Herlant. Theia: Distilling diverse vision foundation models for robot learning. In *Conference on Robot Learning (CoRL)*, 2024.

[40] E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012.

[41] A. Mete, H. Xue, A. Wilcox, Y. Chen, and A. Garg. Quest: Self-supervised skill abstractions for learning continuous control. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[42] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition (CVPR)*, 2016.

[43] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

[44] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, R. Howes, et al. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research (TMLR)*, 2024.

[45] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021.

[46] P. Sermanet, K. Xu, and S. Levine. Unsupervised perceptual rewards for imitation learning. In *Robotics: Science and Systems (RSS)*, 2017.

[47] G. Zhou, H. Pan, Y. LeCun, and L. Pinto. Dino-wm: World models on pre-trained visual features enable zero-shot planning. *arXiv preprint arXiv:2411.04983*, 2025.

[48] J. Zhou, C. Wei, H. Wang, W. Shen, C. Xie, A. Yuille, and T. Kong. ibot: Image bert pre-training with online tokenizer. In *International Conference on Learning Representations (ICLR)*, 2022.

[49] X. Chen, H. Fan, R. Girshick, and K. He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[50] X. Wang, R. Zhang, C. Shen, T. Kong, and L. Li. Dense contrastive learning for self-supervised visual pre-training. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[51] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[52] A. Bardes, J. Ponce, and Y. LeCun. Vicregl: Self-supervised learning of local visual features. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[53] J. Xu and X. Wang. Rethinking self-supervised correspondence learning: A video frame-level similarity perspective. In *International Conference on Computer Vision (ICCV)*, 2021.

[54] S. Dasari, M. K. Srirama, U. Jain, and A. Gupta. An unbiased look at datasets for visuo-motor pre-training. In *Conference on Robot Learning (CoRL)*, 2023.

[55] T. Ridnik, E. Ben-Baruch, A. Noy, and L. Zelnik-Manor. Imagenet-21k pretraining for the masses. In *Advances in Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021.

[56] P. Sundaresan, J. Grannen, B. Thananjeyan, A. Balakrishna, M. Laskey, K. Stone, J. E. Gonzalez, and K. Goldberg. Learning rope manipulation policies using dense object descriptors trained on synthetic depth data. In *International Conference on Robotics and Automation (ICRA)*, 2020.

[57] A. Ganapathi, P. Sundaresan, B. Thananjeyan, A. Balakrishna, D. Seita, J. Grannen, M. Hwang, R. Hoque, J. E. Gonzalez, N. Jamali, et al. Learning dense visual correspondences in simulation to smooth and fold real fabrics. In *International Conference on Robotics and Automation (ICRA)*, 2021.

[58] L. Manuelli, Y. Li, P. Florence, and R. Tedrake. Keypoints into the future: Self-supervised correspondence in model-based reinforcement learning. In *Conference on Robot Learning (CoRL)*, 2021.

[59] P. Florence, L. Manuelli, and R. Tedrake. Self-supervised correspondence in visuomotor policy learning. In *Robotics and Automation Letters (RA-L)*, 2019.

[60] P. R. Florence, L. Manuelli, and R. Tedrake. Dense object nets: Learning dense visual object descriptors by and for robotic manipulation. In *Conference on Robot Learning (CoRL)*, 2018.

[61] L. Yen-Chen, P. Florence, J. T. Barron, T.-Y. Lin, A. Rodriguez, and P. Isola. NeRF-Supervision: Learning dense object descriptors from neural radiance fields. In *International Conference on Robotics and Automation (ICRA)*, 2022.

[62] Y. Wang, Z. Li, M. Zhang, K. Driggs-Campbell, J. Wu, L. Fei-Fei, and Y. Li. D^3fields: Dynamic 3d descriptor fields for zero-shot generalizable robotic manipulation. In *Conference on Robot Learning (CoRL)*, 2024.

[63] N. Tsagkas, O. Mac Aodha, and C. X. Lu. Vl-fields: Towards language-grounded neural implicit spatial representations. In *International Conference on Robotics and Automation Workshops (ICRA)*, 2023.

[64] N. M. M. Shafiullah, C. Paxton, L. Pinto, S. Chintala, and A. Szlam. Clip-fields: Weakly supervised semantic fields for robotic memory. In *Robotics: Science and Systems (RSS)*, 2023.

[65] A. Sochopoulos, M. Gienger, and S. Vijayakumar. Learning deep dynamical systems using stable neural odes. In *International Conference on Intelligent Robots and Systems (IROS)*, 2024.

[66] P. Florence, C. Lynch, A. Zeng, O. Ramirez, A. Wahid, L. Downs, A. Wong, J. Lee, I. Mordatch, and J. Tompson. Implicit behavioral cloning. *Conference on Robot Learning (CoRL)*, 2021.

[67] X. Li, M. Liu, H. Zhang, C. Yu, J. Xu, H. Wu, C. Cheang, Y. Jing, W. Zhang, H. Liu, H. Li, and T. Kong. Vision-language foundation models as effective robot imitators. In *International Conference on Learning Representations (ICLR)*, 2024.

[68] G. Gupta, K. Yadav, Y. Gal, D. Batra, Z. Kira, C. Lu, and T. G. Rudner. Pre-trained text-to-image diffusion models are versatile representation learners for control. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[69] Y. J. Ma, V. Kumar, A. Zhang, O. Bastani, and D. Jayaraman. Liv: language-image representations and rewards for robotic control. In *International Conference on Machine Learning (ICML)*, 2023.

[70] S. Karamcheti, S. Nair, A. S. Chen, T. Kollar, C. Finn, D. Sadigh, and P. Liang. Language-driven representation learning for robotics. In *Robotics: Science and Systems (RSS)*, 2023.

[71] Y. Zhou, S. Sonawani, M. Phielipp, H. Ben Amor, and S. Stepputtis. Learning modular language-conditioned robot policies through attention. *Autonomous Robots*, 2023.

[72] Y. Zhou, S. Sonawani, M. Phielipp, S. Stepputtis, and H. Amor. Modularity through attention: Efficient training and transfer of language-conditioned policies for robot manipulation. In *Conference on Robot Learning (CoRL)*, 2023.

[73] G. Bertasius, H. Wang, and L. Torresani. Is space-time attention all you need for video understanding? In *International Conference on Machine Learning (ICML)*, 2021.

[74] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid. Vivit: A video vision transformer. In *International Conference on Computer Vision (ICCV)*, 2021.

[75] Z. Tong, Y. Song, J. Wang, and L. Wang. VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Appendix

A   Evaluated PVRs

In this section we describe the PVRs evaluated in our experiments. Table A1 summarises the architecture, training objective, and pre-training dataset for each PVR, along with its highlight that motivates our evaluation.

Model	Architecture	Training Objective	Dataset (Size)	Highlight	🤖
MAE [36]	ViT-B/16	Masked Image Modeling	ImageNet (1.2M)	Pioneered MIM model	✘
VC-1 [8]	ViT-B/16	Masked Image Modeling	Ego4D+MNI (5.6M†)	MIM for robot learning	✔
DINOv1 [10]	ViT-B/16	Self Distillation	ImageNet (1.2M)	Seminal self distillation-based SSL model	✘
iBOT [48]	ViT-B/16	Masked Image Modeling + Self Distillation	ImageNet (14M)	Competitive SSL model	✘
DINOv2 [44]	ViT-B/14	Masked Image Modeling + Self Distillation	LVD (142M)	State-of-the-art SSL model	✘
ViT [43]	ViT-B/16	Image Classification	ImageNet (14M)	Earliest Vision Transformer	✘
CLIP [45]	ViT-B/16	Vision-Language Contrastive	LAION (2B)	Widely used VLM	✘
MoCov2 [49]	ResNet-50	Contrastive	ImageNet (1.2M)	Seminal constrastive learning model	✘
DenseCL [50]	ResNet-50	Contrastive (local)	ImageNet (1.2M)	Localised SSL representation	✘
SwAV [51]	ResNet-50	Cluster Assignment Prediction	ImageNet (1.2M)	Representative cluster-based SSL model	✘
VICRegL [52]	ResNet-50	VICReg (global and local)	ImageNet (1.2M)	SSL at both global and local levels	✘
VFS [53]	ResNet-50	Self Distillation (video-based)	Kinetics (240K*)	Representative video frame-based SSL model	✘
VIP [28]	ResNet-50	Goal-conditioned Value Function Learning	Ego4D (5M†)	Reward-oriented representation for robotics	✔
R3M [27]	ResNet-50	Time Contrastive and Language Alignment	Ego4D (5M†)	Temporal-aware representation for robotics	✔

Table A1: Summary of the PVRs evaluated in our experiments, and their respective highlights that motivate our selection. The dataset size denotes the number of images unless specified otherwise. † The number of frames extracted from videos. * The number of videos. 🤖 Pre-trained for robotics.

A.1   The Importance of Training Datasets

The dataset(s) used for pre-training plays a pivotal role in PVRs. While it was hypothesised that pre-training with video data featuring egocentric human-object interaction would be highly effective for learning features suitable for robot learning (due to their emphasis on object manipulation), research indicates that the diversity of images within the dataset is a more critical factor in successful robot learning [54, 8]. Indeed, PVRs pre-trained on static datasets such as ImageNet [55] have demonstrated competitive performance, underscoring the importance of dataset variability over modality.

B   On the use of PVRs in Downstream Robotics Tasks

As discussed in Section 1, while the application of pre-trained visual representations (PVRs) in visuo-motor policy learning is still nascent, these models have proven instrumental in other downstream robotics tasks.

For instance, in manipulation tasks utilizing dense visual descriptors without a learnable policy component, the field has transitioned from training these features from scratch [56, 57, 58, 59, 60, 61] to leveraging features extracted from vision foundation models such as DINO [10, 44] and CLIP [45]. A common strategy involves integrating these features into 3D spatial representations, as exemplified by works like F3RM [5], D^{3}Fields [62], and Click2Grasp [4]. Similarly, PVRs have significantly contributed to semantic mapping, particularly when incorporating language components, as demonstrated in VL-Fields [63] and CLIP-Fields [64].

However, PVRs have yet to gain traction in robot learning frameworks. The foundational study by [13] shows that diffusion policies conditioned on R3M [27] features perform worse than those conditioned on visual encoders trained end-to-end. Notably, policies trained with PVRs tend to produce jittery actions and are prone to getting stuck, which may be linked to an incorrect assumption of the Markov property. Consequently, end-to-end training of visual encoders alongside the policy remains the preferred approach (*e.g.,* [65, 66]).

Finally, an interesting side that is out of the scope of this research concerns the use of other modalities in robot learning. More specifically, a group of recent works has shifted from PVRs that utilise only image data but also language [67, 68, 69, 70].

C   Temporal Encoding

Figs. A1 and A2 extend the methodology from Fig. 1 to real-world data from [13, 71, 72]. As shown, similar issues arise, such as the temporal entanglement of features. While testing our approach on a real robot remains part of our future work, these results provide evidence that the challenges identified in simulation persist in real-world scenarios. Although it is well established that simulations like MetaWorld serve as a reliable proxy for real-world performance [18], these figures underscore the significance of our findings and the necessity of our proposed solutions.

C.1   On the Quality of Simulation Data

Figure A1: Visualisation of PCA results for the features extracted from frames of an expert demonstration trajectory from **the real-world** in the Push-T task [13] across the studied PVRs. In this example, we see mostly a cluster forming when the robot has pushed the T block to its outline and makes very small adjustments.
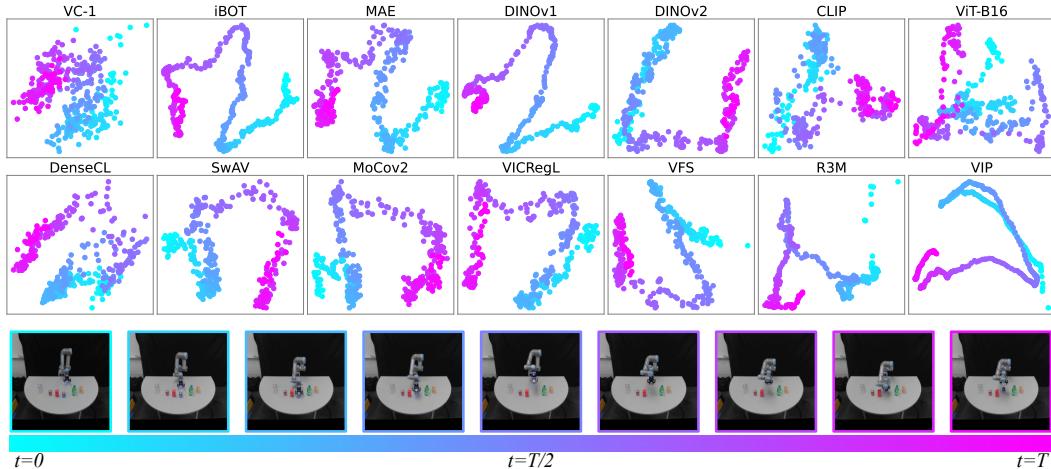

Figure A2: Visualisation of PCA results for the features extracted from frames of an expert demonstration trajectory from **the real-world** in the ASU Table [71, 72] task across the studied PVRs. In this example, entanglement occurs mostly from the fact that the robot picks up the can, rotates it by 90 degrees and then places it in the exact same spot.

C.2 TE Per Task Results

In Table A2, we provide the IQM success rate of policies trained with each PVR-task pair, using the raw features extracted from the PVR, the features temporally augmented with the FLARE method and finally, the features temporally augmented with TE (ours). The results indicate that incorporating TE significantly improves performance compared to using no temporal information. This is supported by both the Wilcoxon test ($p < 10^{-30}$) and paired t-test ($p < 10^{-26}$). The data is not normally distributed. Additionally, TE also outperforms FLARE, with statistically significant differences observed in both the Wilcoxon test ($p \approx 4.38 \times 10^{-5}$) and paired t-test ($p \approx 1.47 \times 10^{-4}$).

	T	DINOv2	DINOv1	MAE	CLIP	ViT	iBot	VC1	MoCov2	SWAV	VIP	DenseCL	R3M	VFS	VICRegL
Bin Picking	−	43.3 ± 0.5	33.3 ± 2.4	50.0 ± 4.3	56.3 ± 1.9	26.7 ± 5.2	41.7 ± 3.3	52.0 ± 2.9	56.7 ± 7.4	54.3 ± 3.3	32.7 ± 3.3	40.7 ± 3.4	40.3 ± 5.4	51.0 ± 4.1	46.0 ± 7.8
	▽	28.3 ± 3.1	35.3 ± 2.4	18.7 ± 2.1	46.7 ± 4.8	32.7 ± 5.2	22.0 ± 2.2	45.0 ± 4.9	38.3 ± 9.6	42.3 ± 0.5	37.7 ± 7.1	29.7 ± 1.7	49.7 ± 6.5	62.3 ± 3.1	44.7 ± 10.4
	◇	66.3 ± 5.2	53.7 ± 3.4	53.7 ± 0.9	71.0 ± 2.2	56.0 ± 2.4	53.3 ± 4.9	53.7 ± 9.9	65.7 ± 3.8	63.7 ± 4.6	51.7 ± 2.1	40.0 ± 8.0	77.0 ± 6.7	79.7 ± 0.9	68.7 ± 0.5
Disassemble	−	85.7 ± 0.5	77.3 ± 0.5	35.7 ± 1.9	76.7 ± 4.2	81.0 ± 2.2	90.3 ± 3.1	74.7 ± 2.9	92.3 ± 1.2	91.3 ± 0.5	85.3 ± 2.5	87.3 ± 2.9	83.0 ± 3.6	80.0 ± 1.6	89.0 ± 0.8
	▽	91.7 ± 1.7	88.7 ± 4.0	76.7 ± 2.5	72.7 ± 3.4	82.0 ± 1.4	89.0 ± 2.4	84.7 ± 3.7	91.3 ± 0.5	91.0 ± 2.2	24.0 ± 7.3	87.3 ± 1.7	91.3 ± 3.9	92.0 ± 0.8	89.7 ± 2.5
	◇	93.7 ± 0.9	91.0 ± 1.6	92.0 ± 2.2	83.3 ± 2.9	90.7 ± 1.7	95.3 ± 2.1	77.3 ± 0.5	92.0 ± 0.8	94.3 ± 0.9	91.0 ± 2.2	90.0 ± 0.8	91.0 ± 2.9	93.7 ± 2.5	93.0 ± 1.6
Coffee Pull	−	46.0 ± 3.6	53.3 ± 2.1	42.3 ± 3.3	42.0 ± 0.8	49.3 ± 1.2	55.7 ± 0.5	19.3 ± 7.1	72.3 ± 1.9	72.0 ± 0.8	62.3 ± 1.7	89.0 ± 2.4	89.3 ± 3.8	82.7 ± 5.0
	▽	47.3 ± 5.4	50.7 ± 1.7	63.0 ± 5.0	50.3 ± 2.1	49.0 ± 1.6	57.0 ± 1.6	60.3 ± 4.2	77.0 ± 0.8	77.0 ± 2.2	88.3 ± 1.7	59.3 ± 0.5	99.0 ± 0.8	86.3 ± 3.3	74.0 ± 2.2
	◇	56.0 ± 3.3	54.0 ± 2.4	47.7 ± 4.7	51.3 ± 3.3	45.0 ± 1.4	58.0 ± 2.9	61.0 ± 3.7	72.0 ± 2.4	77.0 ± 3.6	99.0 ± 0.8	63.0 ± 3.7	93.7 ± 2.6	92.7 ± 2.6	79.3 ± 2.5
Shelf Place	−	9.0 ± 4.2	6.0 ± 2.2	5.0 ± 0.8	10.3 ± 0.9	1.3 ± 1.9	4.0 ± 0.0	2.7 ± 1.7	17.0 ± 4.5	28.7 ± 1.2	28.7 ± 4.5	8.0 ± 0.8	39.7 ± 4.8	20.3 ± 3.3	20.7 ± 4.2
	▽	8.0 ± 0.8	6.7 ± 3.1	8.7 ± 1.2	9.3 ± 0.5	4.3 ± 1.9	3.7 ± 1.9	1.7 ± 0.5	22.7 ± 1.7	23.3 ± 3.4	22.0 ± 2.9	15.0 ± 4.2	43.7 ± 3.4	21.0 ± 1.6	27.7 ± 2.6
	◇	3.7 ± 2.1	5.0 ± 1.4	4.7 ± 2.9	12.3 ± 1.7	6.7 ± 2.5	4.0 ± 0.8	4.3 ± 2.1	20.7 ± 4.2	33.7 ± 5.2	22.0 ± 2.9	12.3 ± 2.1	33.3 ± 1.2	26.7 ± 2.9	28.0 ± 0.8
Peg Insert Side	−	30.7 ± 1.2	23.7 ± 2.1	23.0 ± 1.4	18.7 ± 3.7	4.3 ± 1.2	20.7 ± 3.4	22.0 ± 1.4	34.0 ± 0.8	48.3 ± 3.4	34.0 ± 2.2	32.3 ± 3.7	59.7 ± 0.5	38.0 ± 5.7	38.7 ± 2.1
	▽	24.3 ± 3.8	33.0 ± 5.4	28.3 ± 4.7	13.7 ± 4.9	12.0 ± 2.8	25.3 ± 1.7	17.0 ± 2.4	35.7 ± 3.7	39.3 ± 6.6	44.7 ± 6.3	23.3 ± 2.4	34.7 ± 5.2	35.7 ± 0.9	43.7 ± 3.4
	◇	30.7 ± 0.9	31.3 ± 2.4	27.3 ± 2.9	18.3 ± 0.5	25.7 ± 1.7	11.0 ± 2.8	8.7 ± 0.5	50.7 ± 0.5	37.3 ± 1.7	74.0 ± 0.8	40.3 ± 3.3	69.0 ± 0.8	50.3 ± 3.4	44.3 ± 3.4
Box Close	−	44.0 ± 2.2	59.3 ± 5.4	63.0 ± 6.7	46.3 ± 5.4	57.7 ± 7.4	61.7 ± 1.7	56.3 ± 6.1	56.3 ± 5.3	53.0 ± 2.2	48.7 ± 6.2	59.7 ± 4.5	49.0 ± 6.2	57.3 ± 1.2	61.3 ± 4.5
	▽	45.0 ± 0.8	61.0 ± 1.6	64.0 ± 7.8	55.0 ± 1.6	54.0 ± 2.2	64.7 ± 5.6	60.3 ± 2.1	58.7 ± 5.4	57.7 ± 4.8	33.3 ± 18.8	57.3 ± 5.3	41.3 ± 2.6	62.0 ± 1.6	67.3 ± 2.5
	◇	56.7 ± 1.2	68.3 ± 0.5	57.0 ± 2.2	63.3 ± 5.0	73.3 ± 1.7	61.0 ± 5.0	71.0 ± 2.2	67.0 ± 5.0	61.0 ± 0.8	52.7 ± 0.5	62.3 ± 3.7	68.0 ± 2.2	72.0 ± 3.7	68.3 ± 3.3
Assembly	−	67.3 ± 3.7	61.7 ± 2.9	7.3 ± 0.9	35.3 ± 2.5	55.7 ± 2.5	72.7 ± 3.8	82.3 ± 0.5	90.3 ± 3.1	94.3 ± 0.5	5.0 ± 3.6	97.0 ± 2.2	93.3 ± 1.2	92.3 ± 1.7	93.0 ± 2.4
	▽	24.7 ± 0.5	47.3 ± 2.9	59.7 ± 2.1	34.3 ± 2.5	39.7 ± 3.8	63.3 ± 0.9	50.3 ± 2.1	82.3 ± 4.7	93.0 ± 2.2	35.7 ± 29.8	77.0 ± 2.9	98.3 ± 2.4	92.7 ± 2.1	93.3 ± 1.2
	◇	73.0 ± 4.1	79.3 ± 3.1	66.0 ± 4.3	53.7 ± 1.9	79.3 ± 2.1	61.3 ± 2.5	63.3 ± 3.1	90.3 ± 4.5	91.0 ± 2.8	71.7 ± 8.7	93.7 ± 3.4	100.0 ± 0.0	87.7 ± 3.3	92.7 ± 2.9
Button Press Wall	−	70.3 ± 4.5	76.0 ± 2.9	49.3 ± 1.9	76.7 ± 4.2	62.3 ± 9.0	57.7 ± 1.2	31.0 ± 0.0	57.3 ± 0.9	69.3 ± 2.6	22.3 ± 17.4	52.0 ± 7.9	63.3 ± 3.9	58.7 ± 4.6	51.3 ± 5.4
	▽	59.3 ± 7.0	84.7 ± 2.9	63.3 ± 4.9	67.7 ± 1.2	64.7 ± 1.2	74.0 ± 5.4	65.0 ± 0.8	63.7 ± 7.5	64.0 ± 4.9	65.7 ± 14.8	68.0 ± 2.9	60.3 ± 2.6	62.0 ± 6.5	50.0 ± 1.6
	◇	79.0 ± 4.2	77.3 ± 2.4	85.0 ± 0.0	82.0 ± 2.8	69.0 ± 4.3	74.3 ± 0.9	63.0 ± 3.7	77.7 ± 8.7	64.7 ± 3.4	6.3 ± 2.5	83.3 ± 4.8	54.7 ± 1.2	58.0 ± 5.7	67.0 ± 12.7
Pick Place Wall	−	39.7 ± 7.4	29.3 ± 4.5	41.0 ± 4.5	36.0 ± 5.7	59.3 ± 7.3	25.7 ± 1.7	49.7 ± 7.1	84.0 ± 1.6	69.3 ± 1.7	54.0 ± 15.7	61.7 ± 5.4	67.0 ± 4.9	80.7 ± 0.5	76.3 ± 2.4
	▽	47.0 ± 6.2	69.3 ± 5.3	52.0 ± 4.2	49.0 ± 2.2	64.3 ± 3.3	58.7 ± 2.1	44.3 ± 2.6	88.7 ± 2.4	89.0 ± 0.8	86.3 ± 2.5	59.0 ± 9.6	95.3 ± 0.5	90.0 ± 0.8	95.0 ± 2.4
	◇	24.7 ± 2.5	59.0 ± 7.3	26.7 ± 4.5	29.7 ± 4.2	40.0 ± 2.2	24.3 ± 0.9	26.0 ± 4.3	79.7 ± 0.9	62.0 ± 3.7	86.7 ± 2.6	49.7 ± 2.6	66.0 ± 0.8	83.3 ± 5.2	73.3 ± 4.6
Drawer Open	−	90.7 ± 1.2	92.0 ± 0.0	92.7 ± 1.2	87.7 ± 0.9	89.3 ± 0.5	96.3 ± 1.2	95.7 ± 1.7	93.3 ± 1.9	93.3 ± 1.9	95.7 ± 1.7	100.0 ± 0.0	90.3 ± 1.2	95.0 ± 0.8	92.0 ± 0.0
	▽	93.0 ± 1.4	93.3 ± 1.9	94.0 ± 1.6	90.3 ± 1.2	89.3 ± 0.5	99.7 ± 0.5	100.0 ± 0.0	100.0 ± 0.0	95.0 ± 0.8	91.7 ± 0.5	100.0 ± 0.0	100.0 ± 0.0	95.0 ± 0.8	92.0 ± 0.0
	◇	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	99.3 ± 0.9	99.3 ± 0.9	100.0 ± 0.0	100.0 ± 0.0	95.7 ± 0.5	100.0 ± 0.0	95.7 ± 1.7	98.0 ± 0.8	100.0 ± 0.0	100.0 ± 0.0	94.7 ± 0.9
Average	−	52.7	51.2	40.9	48.6	48.7	52.6	48.6	65.3	67.4	50.6	60.1	67.5	66.3	65.1
	▽	46.9	57.0	52.8	48.9	49.2	55.7	52.9	65.3	66.8	54.7	57.6	71.4	69.9	67.7
	◇	58.4	61.9	56.0	56.4	58.5	54.2	52.8	71.2	68.5	65.1	63.3	75.3	74.4	70.9

Table A2: Evaluation of trained policies across 10 tasks, utilising features from 14 different PVRs. Results are reported without any temporal augmentation (T: −), with the FLARE method (T: ▽) and with TE of the timestep (T: ◇). We mark with green the temporal encoding results that outperformed the other two methods. Similarly, with red for the no augmentation case and yellow for FLARE.

C.3 Tuning Temporal Encoding Hyperparameters

We fine-tune TE's scale and dimensionality using 4 PVRs (MAE, DINOv1, SwAV, R3M) and 4 MetaWorld tasks (Bin Picking, Coffee Pull, Disassemble, Pick and Place). Table A3 shows the percentage of PVR-task pairs benefiting from TE across different hyperparameters, with $D = 64$, $s = 100$ yielding the most improvements. As seen in Table A4, this setting also achieves one of the highest average performance gains.

Dims	10	100	1000
64	56.25%	84.38%	75.00%
128	81.25%	75.00%	75.00%
256	75.00%	81.25%	81.25%

Dims	10	100	1000
64	11.26 ± 6.66	11.82 ± 5.72	9.08 ± 4.43
128	8.51 ± 7.61	11.97 ± 4.79	11.03 ± 5.39
256	8.89 ± 4.84	11.38 ± 6.17	11.82 ± 6.73

Table A3: Percentage of PVR-Task pairs where TE led to performance increase.

Table A4: Average boost (i.e., for the PVR-Task pairs that benefited from TE) values for each scale and dimension combination.

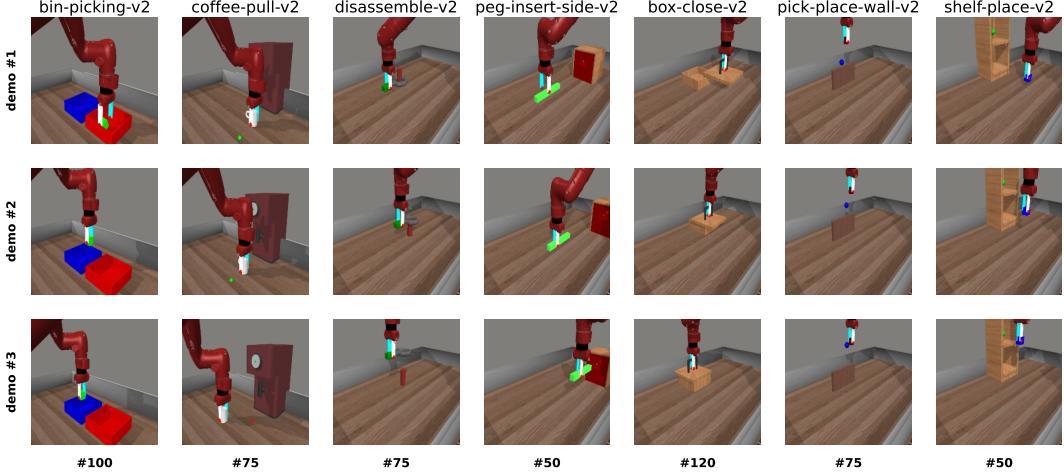## C.4 MetaWorld Temporal Variability of Expert Demos

Figure A3: Illustration of temporal variability in MetaWorld expert demonstrations. Each column represents a different task, with each row showing a frame captured at the same time-step across three separate demonstrations. The lack of perfect synchronisation between demonstrations highlights variability in task progression.

C.5 Video Encoders as PVRs

Even though in this work we investigate in the context of visuo-motor policy learning the role of PVRs that have been trained with massive image datasets, we also briefly consider pre-trained video encoders as an alternative solution. Our motivation stems from the fact that such models, which have been trained using objective functions that aim to encode in the model strong temporal perception, could potential naturally resolve the identified issue of temporal entanglement. For this purpose, we deploy three powerful video encoders (TimeSformer [73], ViViT [74] and VideoMAE [75]) that have all been trained with the Kinetics-400 dataset and utilise the ViT-B/16 backbone.

In the new experiments we preserve our methodology apart from the way the frame input stream is processed. In the case of video encoders, for both training and evaluating, a frame buffer is created of length N_f, which has the most recent frame, followed by the $N_f - 1$ previous ones. Until the number of available frames become equal to N_f, the buffer is padded, by repeating the oldest frame. We set N_f to match the frame history length of each encoder [1].

	ViT-B/16	TimeSformer	VideoMAE	ViViT
N_f	1	8	16	32
T_p	$\approx 0.025s$	$\approx 0.145s$	$\approx 0.265s$	$\approx 0.550s$
Video-PVR	–	56.9%	45.5%	18.8%
Video-PVR + TE	–	62.4%	44.8%	24.9%

Table A5: Comparison of inference time and input frames across three video encoders versus the vanilla ViT-B/16, the base for all ViT-based PVRs except DINOv2 (patch size 14), which processes a single frame.

Table A5 summarises two important aspects of pre-trained video encoders. First, we measure the average inference time of each encoder and compare it against the time it takes to process a single frame for the same backbone (*i.e.,* ViT-B/16), which is the one utilised by other PVRs in

---

[1]All encoder preprocessing modules and model inference times were measured using code from `huggingface.co/docs/transformers` and tested on a NVIDIA GeForce RTX 4090 GPU with 24GB VRAM, using batches of size 25.

our experiments. It is not a surprise that the larger N_f is, the slower inference gets. An interesting find concerns the average success rate itself on the MetaWorld tasks, which seems to be negatively correlated with the number of frames in the buffer. This counter-intuitive result aligns with the findings of [13], regarding the length of the observation horizon, where the performance would decline as the length increased.

C.6 Testing TE on a Causal Transformer

To further test the usefulness and effectiveness of TE, we deploy it on a policy architecture that implicitly models temporal aspects of the rollout. We use a 1-layer causal transformer (CT) that processes features extracted from R3M. This increases the overall capacity of the model by 25,186,304 trainable parameters (note that this number would be significantly smaller if we were working with a ViT-B/16 PVR).

We use context and action chunking of length 12 and train for 0.8M steps on four MetaWorld tasks. We incorporate the temporal embeddings on the output of the CT, hypothesising that the entanglement remains. Indeed, Table A6 reveals that augmenting the input feature space with a temporal component and a sense of task completion benefits greatly the policy in successfully completing a task, leading to an average task success rate boost of +20%.

	Peg Insert	Bin Picking	Disassemble	Coffee Pull	Average
CT	42%	80%	54%	96%	68.0%
CT + TE	62%	90%	93%	100%	86.3%

Table A6: Success rate on four different tasks of a Causal Transformer trained with and without TE.

C.7 PVR vs PVR+TE vs PVR+AFA vs Ours

Fig. A4 validates our full model's significance via an ablation study. We evaluate the contributions of TE and AFA in our approach, comparing it against training the policy network solely with PVR-extracted features. In all cases, TE improves the corresponding policy, regardless of PVR choice and use of AFA. On average, AFA improves on slightly the in-domain performance (*i.e.,* from 63.1% to 66.4%). Nevertheless, the real value of this module is highlighted in Fig. 4.
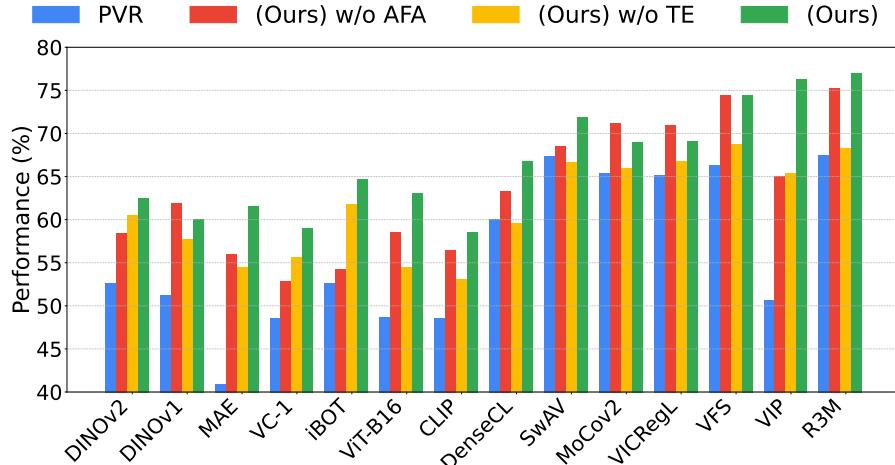

Figure A4: Ablation study on our approach (*i.e.,* PVR+TE+AFA). Results are reported on in-domain scenes.

D Attentive Feature Aggregation

D.1 AFA Per Task Results

In Fig. A5 we provided the results for each PVR+AFA+TE-task pair, for both in-domain scenes and visually altered scenes.
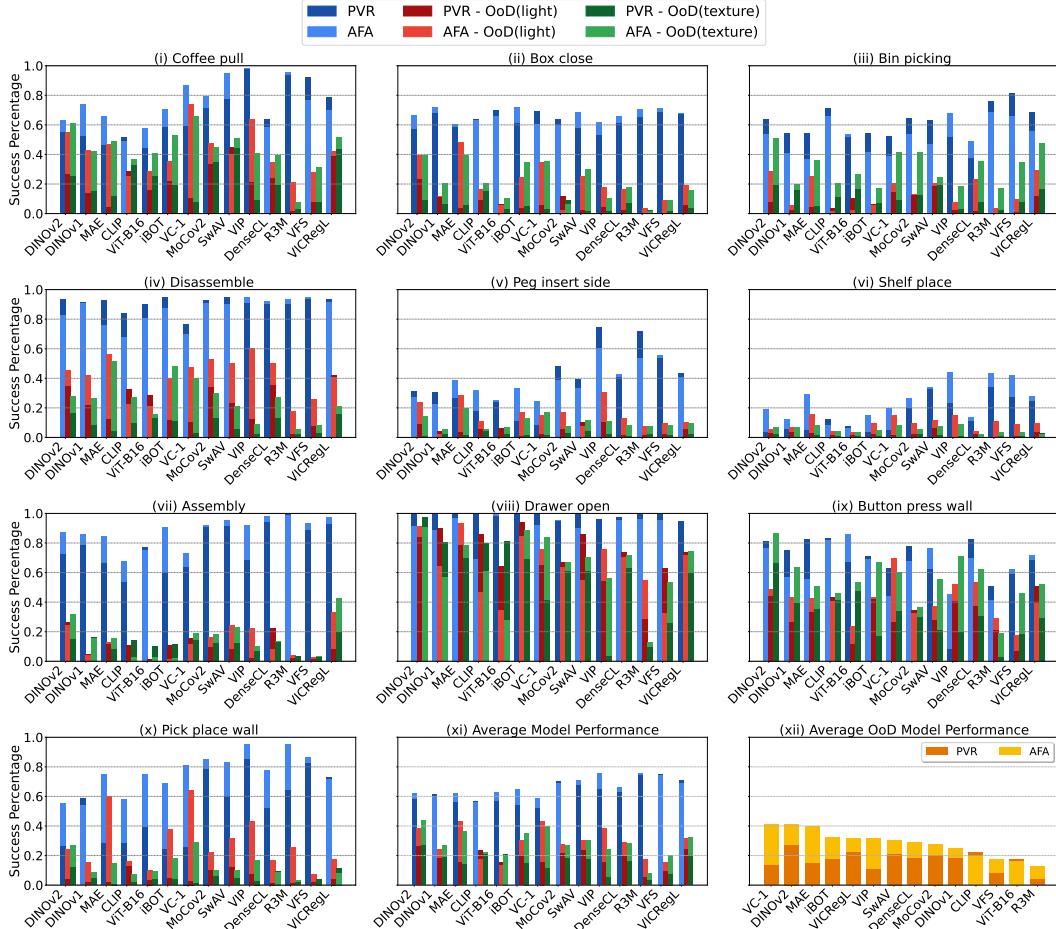

Figure A5: Evaluation of the AFA module in both in-domain and out-of-domain scenarios, including tabletop texture and lighting perturbations. Sub-figures (i)-(x) illustrate policy performance for individual tasks. Sub-figure (xi) presents the average performance across all tasks, while sub-figure (xii) displays the average OOD performance, with PVRs sorted by descending ABC performance.

D.2 Scene Perturbations

To test the robustness of trained policies we visualise modify the scenes in the evaluation either by changing the lighting or by randomly changing the tabletop texture. Note that all policies are evaluated in the same perturbations for fairness.

Randomizing the scene's lighting properties.The brightness of the scene is altered by adjusting the diffuse light components, where each colour channel (red, green, blue) is randomly set to a value between 0.3 and 1.0. The specular highlights are similarly randomized, with lower intensity values ranging from 0.1 to 0.5. Additionally, the position of each light source is varied randomly within a 3D space, spanning horizontal and vertical shifts between -2 and 2 units and height adjustments between

0.5 and 3 units. Lastly, the direction of the lights is randomized, allowing for changes in their angular orientation, with each directional component varying between -1 and 1 for horizontal/vertical angles and up to -1 for downward angles.

Randomizing the Tabletop's texture. In Fig. A6 we provide the textures that we utilised in our experiments, some of which are borrowed from [11]. Some are visually similar to the texture used in the training demos and others are vibrant, with patterns that hold semantic information that could potentially attract the attention of a PVR. Nevertheless, by observing the evaluation rollouts, policies can fail and succeed in both out-of-distribution cases.

Figure A6: Visualisation of the different table textures used in the evaluations of Section 4.3. The additional textures that are not provided by MetaWorld were borrowed from [11].

Figure A7: Visualisation of the 10 tasks used for evaluation. The first row illustrates representative scenes for all tasks, as seen in the frames from the expert demonstrations (in-domain). The second row shows how the scenes are modified by randomly altering the brightness, orientation and position of the light source. Similarly, the third row presents changes to the tabletop texture.

D.3 Ablation: Does AFA's Success Stem from its Increased Capacity?

Does AFA's success stem from its increased capacity? We make the policy network deeper (*i.e.,* from 594,956 to 1,645,580) to roughly match the number of trainable parameters of the AFA (*i.e.,* 1,774,336) by adding more layers. In Fig. A8 we visualise the results both in and out of domain for policies trained with AFA and with the deeper MLP. As is evident, AFA still has the better robustness performance.

D.4 Qualitatively Explaining AFA's Performance Boost

In Fig. A9, we present additional comparisons between the attention heatmaps of the PVRs and their corresponding trained AFAs. These visualizations illustrate how the CLS token attends to different patches compared to the trained query token, offering a general sense of what the models prioritise. While this does not imply that trained AFAs are entirely robust to visual changes in the scene (*e.g.,* note that iBOT+AFA still allocates some attention to patches containing the robot's cast shadow),
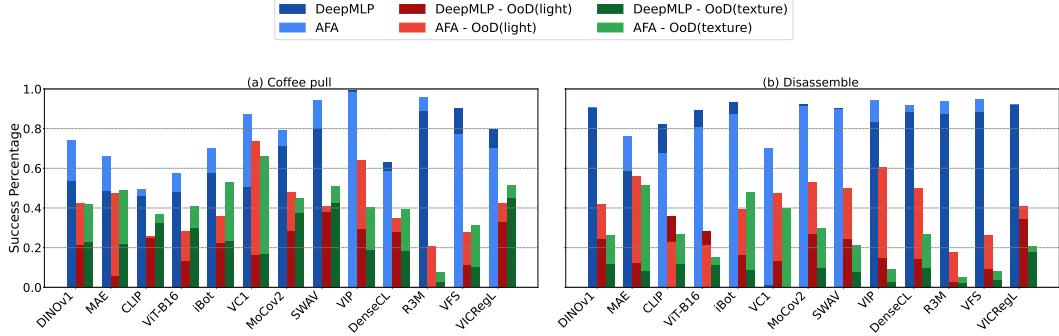
Figure A8: Results from policies trained with either AFA or with a deeper MLP of capacity comparable to that of AFA.

we observe a consistent trend: the attention heatmaps become more focused, particularly on regions relevant to the task.
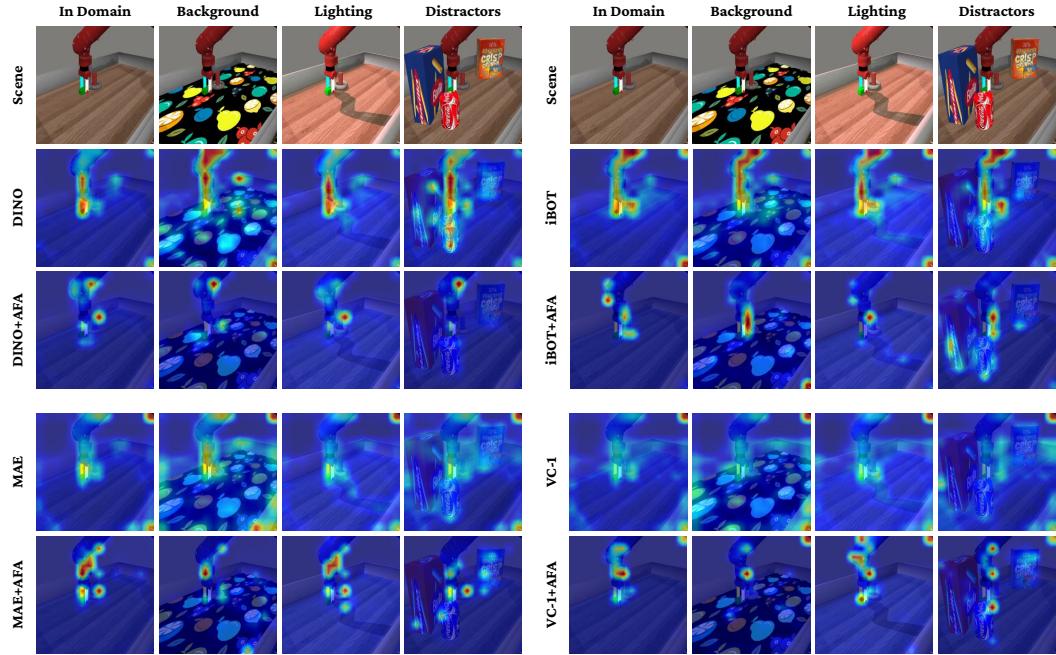

Figure A9: Additional comparisons between PVR and PVR+AFA attention heatmaps.

D.5   Ablating the Pooling Mechanism

In this work, our objective is not to simply turn a stream of tokens into a more compact representation, as is usually the case, but rather learn to automatically identify and preserve task relevant visual cues, while at the same time discarding the unrelated ones. Our experiments provide evidence that AFA increases policy robustness, increasing performance under scene perturbations. Nevertheless, other pooling mechanisms have played an imperative role in visuo-motor policy learning.

Spatial Softmax. One such mechanism is Spatial Softmax, which has been widely used in visuo-motor policy learning [15, 13] for extracting spatially meaningful features from convolutional feature maps. Spatial Softmax operates by interpreting the activation map as a probability distribution over spatial locations, thereby encouraging the network to focus on the most salient regions in the image. This results in a differentiable way to extract expected spatial coordinates of visual features, which can be directly fed into downstream policy networks.

Figure A10: Per task policy success rate for three pooling techniques (AFA-ours, Spatial Softmax [15], TokenLearner [16]) with and without scene visual changes.

Given a feature map $f \in \mathbb{R}^{C \times H \times W}$, where C is the number of channels, and H, W are the height and width respectively, the Spatial Softmax is applied to each channel c as follows:

$$s_{ij}^{(c)} = \frac{\exp(f_{ij}^{(c)})}{\sum\limits_{i'=1}^{H} \sum\limits_{j'=1}^{W} \exp(f_{i'j'}^{(c)})} \tag{1}$$

Using the resulting softmax weights $s_{ij}^{(c)}$, the expected 2D coordinates $(\hat{x}^{(c)}, \hat{y}^{(c)})$ for each channel are computed as:

$$\hat{x}^{(c)} = \sum_{i=1}^{H} \sum_{j=1}^{W} s_{ij}^{(c)} \cdot x_j, \quad \hat{y}^{(c)} = \sum_{i=1}^{H} \sum_{j=1}^{W} s_{ij}^{(c)} \cdot y_i \tag{2}$$

where x_j and y_i denote the horizontal and vertical coordinates normalized to a fixed range (typically $[-1, 1]$).

This pooling technique thus compresses high-dimensional spatial information into a compact and interpretable form, allowing the policy to attend to semantically meaningful visual cues—such as object positions—while remaining end-to-end differentiable. In our context, we compare AFA's learned attention masks to Spatial Softmax in terms of robustness and relevance to task-specific perturbations.

TokenLearner. Another relevant mechanism is TokenLearner [16], which dynamically selects a small number of informative tokens from the full set of visual tokens, thereby reducing computational complexity while preserving task-relevant information. Unlike static pooling operations, Token-Learner introduces learnable spatial attention maps that are conditioned on the input, allowing the model to adaptively select visual regions of interest.

Given an input feature map $F \in \mathbb{R}^{H \times W \times D}$, TokenLearner computes M attention maps $A^{(m)} \in \mathbb{R}^{H \times W}$ using a lightweight module, typically an MLP or convolutional layers followed by a softmax over the spatial dimensions:

$$A^{(m)} = \text{softmax}(\phi_m(F)), \quad \text{for } m = 1, \ldots, M \tag{3}$$

where ϕ_m denotes the attention function for the m-th token.

Each learned token $T^{(m)} \in \mathbb{R}^D$ is then computed as the weighted sum over spatial locations:

$$T^{(m)} = \sum_{i=1}^{H} \sum_{j=1}^{W} A_{ij}^{(m)} \cdot F_{ij} \qquad (4)$$

The resulting set of tokens $\{T^{(1)}, \ldots, T^{(M)}\} \in \mathbb{R}^{M \times D}$ serves as a compact yet informative representation that can be passed to a Transformer encoder or a policy network.

In our framework, we compare AFA to TokenLearner by evaluating their capacity to isolate and preserve task-relevant spatial cues. While both methods introduce adaptive pooling, AFA further encourages robustness under spatial perturbations by integrating attention masks optimized directly for downstream control performance.

Ablation. To isolate the contribution of AFA, we compare it against Spatial Softmax and Token-Learner. To make the comparison fair, in TokenLearner we learn to extract a single token, both because in AFA we use a single trainable query token, but also because the policy head process a single embedding. For Spatial Softmax, we extract a pair of 2D coordinates per feature dimension. Evaluating all tasks in and out of domain with the three pooling techniques using features from the top performing PVR (*i.e.,* VC-1), we can see that AFA is by far the most robust method under visual changes (see Fig. A10), outperforming both methods by approximately 20%. A notable observation is that Spatial Softmax demonstrates on average the best performance, which justifies its use in the vanilla Diffusion Policy [13].

Interpretation. We argue that this result is not surprising. TokenLearner operates on feature level, ignoring the information structure that is encoded per attention head. On the other hand, AFA generates a specific weight for each channel group for each per-patch token. Additionally, Spatial Softmax does not try to favour specific visual cues, but rather compress the image information. As a result, redundant cues that are unrelated to the task remain.

D.6 Real-World Experiments

We evaluated our method in the real-world. We selected a simple planar pushing task, where the goal is for the robot to push a cube inside of a marked outline on the table, matching the colour of each side. The robot can only move on the (x, y) plane to push the cube, using a rod that has been attached to the end-effector. We collected 20 demonstrations via teleoperation with the KUKA IIWA 14 robot and a Realsense D415 camera that observes the workspace. We provide four examples of these demonstrations in Fig. A13.
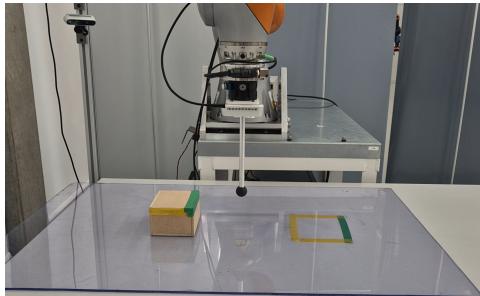
We executed the trained policies using a NVIDIA GeForce RTX 2080, generating a new action at 10Hz. For the real-world experiments we utilised features extracted from DINOv1 [10], which we empirically observed performed better at the particular task compared to other models.

We pre-trained for 0.2M steps a policy model with and without AFA, using the 20 collected demonstrations. Both models had no problem solving the task in-domain (see Fig. A11 (a)). Nevertheless, under visual perturbations, the policy trained with the DINOv1 `cls` token was unable to push the cube within the outline on the table, in many cases performing actions that seemed random. On the other hand, when trained with AFA, using the per patch tokens, the policy exhibited robustness, solving all the tasks with distractors and lighting changes.
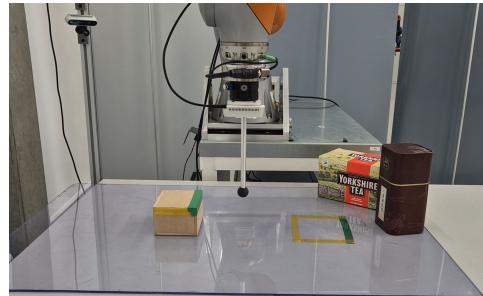
In the real-world experiments we introduced two types of perturbations. First, we modified the scene by adding distractors near the goal, specifically selecting items rich in semantic content that pose

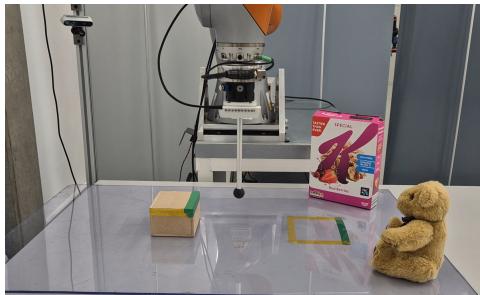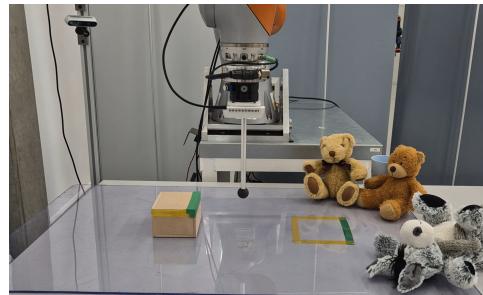|	In-Domain	Obj. Distractors	Lighting Changes	Cube Mod.
PVR	4/5	0/4	0/3	0/1
PVR + AFA	4/5	4/4	3/3	1/1

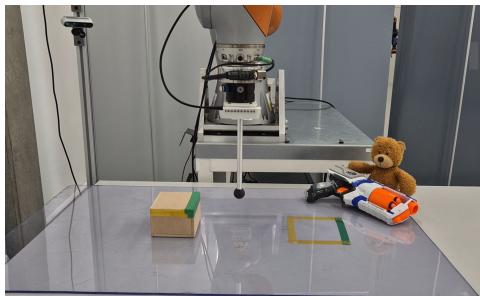Table A7: Real-world success rate.

(a) No visual perturbations.

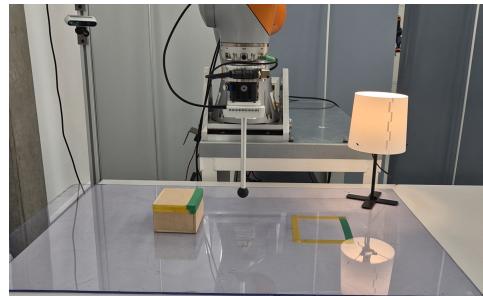(b) Scene with distractors No. 1.
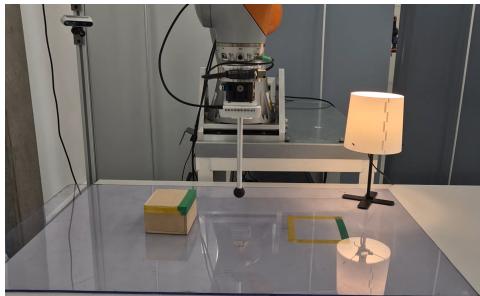
(c) Scene with distractors No. 2.
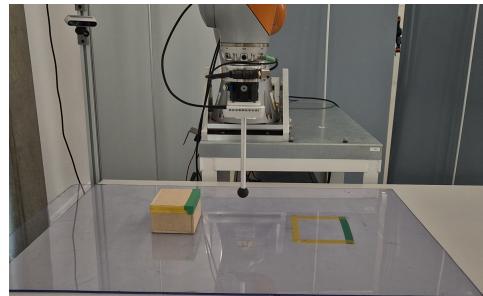
(d) Scene with distractors No. 3.

(e) Scene with distractors No. 4.

(f) Ceiling lights dimmed and desk lamp.

(g) Ceiling lights off and desk lamp.

(h) Ceiling lights off.

Figure A11: **Eight real-world evaluation configurations**. Scene (a) represents the in-domain setting, identical to the expert demonstrations. Scenes (b)–(e) introduce visual distractor objects, while scenes (f)–(h) feature variations in lighting conditions.

strong challenges for PVRs, pre-trained with real-world images (*e.g.,* stuffed toys and groceries), as shown in Fig. A11 (b)-(e). Second, we changed the lighting conditions, both by dimming the ceiling lights and including a light source near the goal, as shown in Fig. A11 (f)-(h). The reflective surface made this visual change even more pronounced. Finally, we added a distractor to the cube itself, thus visually modifying the object of interaction, as presented in Fig. A12. This example is considerably more difficult since the emergency button we placed on top of the cube obstructs important cues (*e.g.,* part of the green tape on the cube), due to the camera position and orientation (see Fig. A13). Nevertheless, when training the policy with AFA, the robot was able to roughly position the cube within the layout. On the other hand, without AFA, the policy was executing almost random motions.
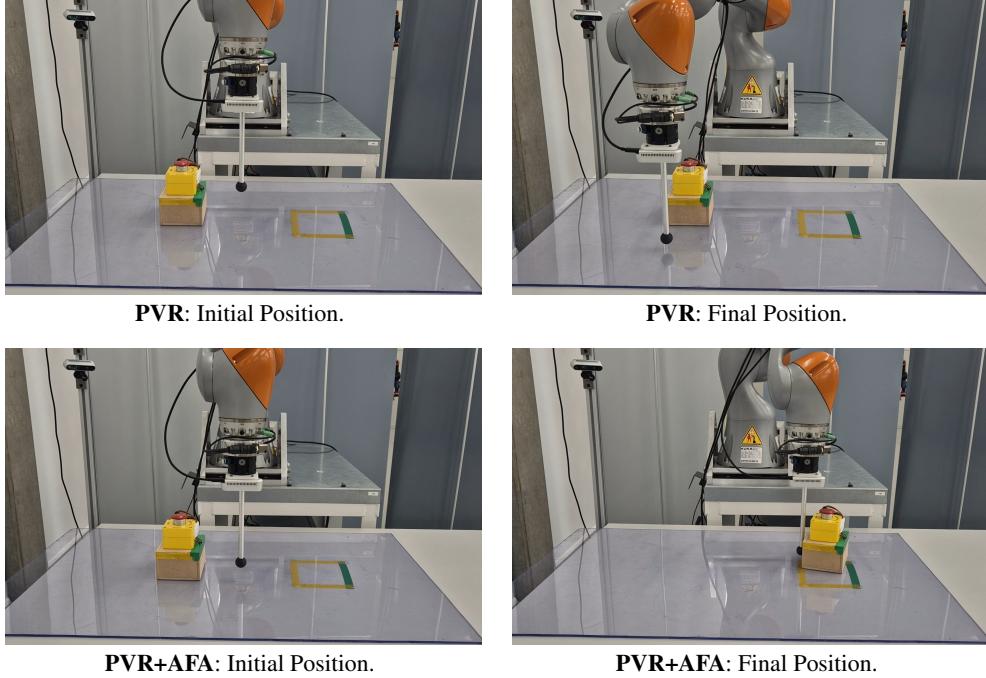

PVR: Initial Position.

PVR: Final Position.

PVR+AFA: Initial Position.

PVR+AFA: Final Position.

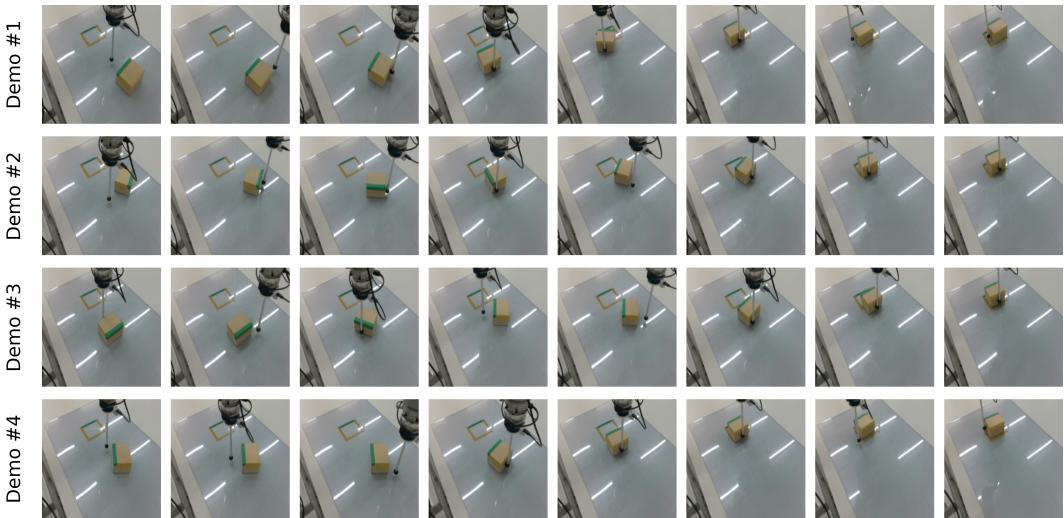Figure A12: **Special OoD case**. Visually modifying the cube itself.

Figure A13: Examples of expert demonstrations for the planar pushing task.