

Appendix A: Performance and Computation Trade-off of Saliency Buffer

Saliency Buffer Implementation Details

- **Image Global Index:** Each image sample is assigned a global index, which is included as additional data returned by the data loader.
- **Random Cropping:** When random cropping (pixel shift) is enabled, cropping randomisers return the cropping parameters, which include the pixel coordinates of the top-left corner and the cropping size. Assuming no additional down-sampling when saving, the buffer maintains the saliency map at its original size before cropping. A saliency map derived from cropped image is padded with zeros in the regions cropped out to match the full image size. Upon retrieval, saliency maps are adjusted to the specified cropping parameters.
- **Pixel Range Conversion:** Each saliency map stored in the buffer is kept as a single-channel 8-bit unsigned integer (UINT8) image to optimise memory use. For augmentation, saliency maps are normalised with pixel values scaled to $[0, 1]$. These values are converted to UINT8 when saved and back to float32 (FLOAT32) within $[0, 1]$ upon retrieval.
- **Saliency Warm-up:** The buffer does not update and returns all-one saliency maps during the first γ epochs (default set to 10), allowing the encoders to develop task-specific knowledge before strong data augmentation is applied. This warm-up strategy is also implemented in the RoboSaGA variants: Random Overlay and Guided-Erase.

Performance and Computation Trade-off

	Lift		Can		Square	
	Buffer	No-Buffer	Buffer	No-Buffer	Buffer	No-Buffer
In-domain	0.98 ± 0.00	0.99 ± 0.01	0.97 ± 0.03	0.96 ± 0.03	0.68 ± 0.06	0.69 ± 0.01
Background	0.89 ± 0.05	0.92 ± 0.03	0.87 ± 0.02	0.85 ± 0.01	0.39 ± 0.08	0.43 ± 0.02
Distractor	0.95 ± 0.04	0.98 ± 0.02	0.71 ± 0.07	0.67 ± 0.03	0.69 ± 0.09	0.67 ± 0.02
<i>mean</i>	0.92 ± 0.05	0.95 ± 0.04	0.79 ± 0.09	0.76 ± 0.10	0.54 ± 0.17	0.55 ± 0.12

Table 4: Performance of RoboSaGA with and without Saliency Buffer under visual domain shifts (BC-MLP).

Buffer No Buffer
0.22s 0.70s

Table 5: Average augmentation time per batch

Here, we compare the performance and computational differences between utilising a saliency buffer and directly computing the saliency. In the RoboSaGA experiments, detailed in Sec. 4.2, 10% of the current batch is sampled for saliency updates and saved in the saliency buffer; the augmentation ratio α is maintained at 50% of the current batch. By contrast, when the buffer is disabled, saliency maps are directly computed for 50% of the batch. The experiments are conducted on the *Lift*, *Can*, and *Square* tasks in simulation using the BC-MLP policy.

As shown in Tab. 4, the average success rates over three tasks for RoboSaGA, with and without the saliency buffer under visual domain shifts, are 0.75 and 0.753, respectively. No significant performance difference is observed. While the use of the saliency buffer does not enhance performance, it significantly reduces the computation time required for saliency extraction to one-third (Tab. 5).

Task	Action Dimension	Observation Dimension	Proprio. Dimension	Max. Steps	Num. of Views	Long Horizon	Num. of Subtasks
Simulation							
<i>Lift</i>	7	$2 \times 84 \times 84$	7	200	400	No	1
<i>Can</i>	7	$2 \times 84 \times 84$	7	200	400	No	2
<i>Square</i>	7	$2 \times 84 \times 84$	7	200	400	No	2
<i>Transport</i>	14	$4 \times 84 \times 84$	7	200	400	Yes	8
Real-world							
<i>Toy</i>	7	$2 \times 84 \times 84$	12	160	–	No	2

Table 6: **Task Summary.** *Action Dimension*: Robot action dimension, including end-effector 6-DOF velocity and parallel gripper width. *Observation Dimension*: Number of views \times Image observation dimension. *Proprio. Dimension*: the dimension of robot proprioceptive states. *Num. of Demos*: Number of demonstrations provided. *Maximum Steps*: Maximum rollout steps in simulation. *Num. of Views*: Number of camera views. *Long-Horizon*: Indicates whether the task requires learning multiple behaviours together. *Num. of Subtasks*: Number of sub-tasks within each task.

Appendix B: Experimental Details

Task Descriptions

In this work, we utilise four simulation tasks from RoboMimic [5] with provided proficient human (PH) demonstrations, and one real-world pick-and-place task collected via proficient human teleoperation. All tasks employ Franka Panda as the manipulator within a 7-dimensional action space, which includes the 6 degrees of freedom for end-effector pose and gripper width. In the simulation, the proprioceptive states include the end-effector pose (rotation is represented by quaternion) and gripper width. The real task utilises a 12-dimensional robot state represented by a flattened SE(3) matrix (last row omitted). Except for the *Transport* task, all tasks use two camera views: a second-person camera and a first-person camera. The *Transport* task features two camera views associated with each manipulator. Task descriptions and details are summarised below and in Tab. 6.

- *Lift*: Grasp and lift a red cube from the table.
- *Can*: Grasp a Coke Can from the left bin (sub-task 1), and place it into the corresponding target bin on its right (sub-task 2).
- *Square*: Grasp the square nut by the handle (sub-task 1) and insert it into a matching square peg (sub-task 2).
- *Transport*:
 - Left arm: Pick the handle of the source bin’s lid (sub-task 1), place the lid in the empty space (sub-task 2), grasp the hammer within the source bin (sub-task 3), and deliver it to the workspace within the right arm’s reach (sub-task 4).
 - Right arm: Pick up a red cube from the target bin (sub-task 5), place it into the trash bin (sub-task 6), take the hammer delivered by the left arm (sub-task 7), and place it into the target bin (sub-task 8).
- *Toy*: Pick up a green squashy toy (sub-task 1) and place it into a yellow cup (sub-task 2). The location of the toy and the cup varies within a $10 \times 10\text{cm}^2$ area.

Policy and Training Details

BC-MLP and *BC-RNN* utilise the default network configurations as specified in RoboMimic [5]. Specifically, each image observation is processed by a ResNet18 [24] followed by a spatial-softmax layer [25]. All observation features are then concatenated with the robot’s proprioceptive states, forming a single feature vector. The flattened states are fed into either an MLP or an RNN. The MLP employs ReLU activations, while the RNN consists of a 2-layer LSTM. The final layer’s

Hyperparameter	BC-MLP	BC-RNN
Learning Rate (LR)	1×10^{-4}	1×10^{-4}
Actor MLP Dimensions	[1024, 1024]	-
RNN Hidden Dimension	-	1000
RNN Sequence Length	-	10
GMM Number of Modes	5	5
Image Encoder	ResNet18	ResNet18
SpatialSoftmax (num-KP)	64	64

Table 7: Hyper-parameters for BC-MLP and BC-RNN.

hidden states are fed into the downstream Gaussian Mixture Model (GMM). As described in RoboMimic [5], during the rollout with GMM policies, the learned standard deviations of each mode are replaced with 1×10^{-4} . Hyper-parameter settings are detailed in Tab. 7.

Diffusion Policy employs the hybrid-CNN architecture as detailed in [2]. It utilises a similar image encoder as described above but replaces BatchNorm layers [26] with GroupNorm [27]. The input horizon, action horizon, and action prediction horizon are set to 2, 8, and 16, respectively. The learning rate is set to 1×10^{-4} .

All networks are trained from scratch with Adam [28] optimiser (learning rate set to 1×10^{-4}). Except for the simple *Lift* task is trained for 200 epochs, all tasks are trained for 600 epochs. 10% of random cropping (i.e., 76×76 for 84×84 inputs) is applied to all images during training and 10% of centre cropping during evaluation.

Out-of-domain Images for RoboSAGA



Figure 6: Examples of out-of-domain images for data augmentation

RoboSAGA utilizes approximately 6,000 out-of-domain images for augmentation, comprising 5,000 real-world images from MSCOCO [23] and 1,000 synthetic images featuring plain, gradient, grid, chess, and Perlin patterns (Fig. 6). All images are subject to random rotations and brightness adjustments.

Backgrounds and Distractors for Evaluation

In the *simulation*, as illustrated in Fig. 7, table textures are derived from images of textiles and patterns. Floor textures utilise common materials such as tiles and wooden flooring, while wall textures are selected from both indoor and outdoor scenes. Distractors, including items like a Coke can, bottle, cereal, bread, and lemon, form the default set. Some distractors may be removed to prevent duplicating task-specific objects. For example, the Coke can is excluded from the *Can* task. Distractors are strategically placed to minimise collisions with the manipulator and targets, although collisions are still possible. Each selected distractor has a 50% chance of appearing.

In the **real-world** setting, as shown in Fig. 8, background shuffling is achieved by varying the combinations of textiles drawn from the default textile set. These textiles sufficiently cover the



Figure 7: Examples of textures and distractors in simulation evaluation

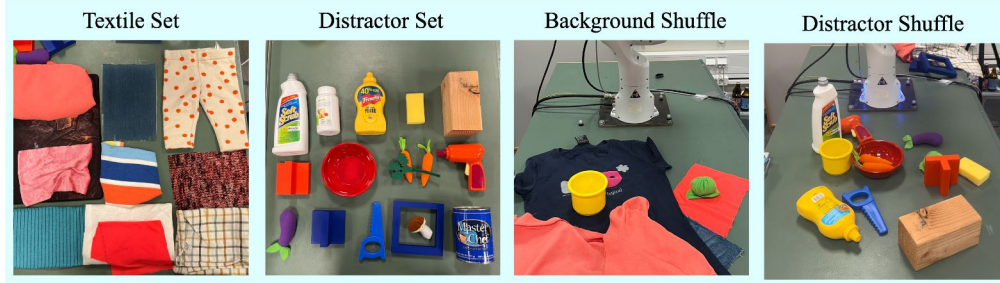


Figure 8: Examples of textiles and distractors in real-world evaluation

field of view for both the second-person camera and the eye-in-hand camera, with examples of the respective fields of view illustrated in Fig. 5. Distractors are also drawn from the default distractor set, forming cluttered arrangements.

Appendix C: Full Tables for Simulated Experiments

		BC				BC-RNN			
		None	Overlay	SODA	SaGA	None	Overlay	SODA	SaGA
Domain									
Lift	In-domain	0.97 \pm 0.01	0.99 \pm 0.01	1.00 \pm 0.00	0.98 \pm 0.00	0.99 \pm 0.01	1.00 \pm 0.00	0.95 \pm 0.02	0.99 \pm 0.02
	Background	0.00 \pm 0.00	0.87 \pm 0.05	0.80 \pm 0.04	0.89 \pm 0.05	0.02 \pm 0.03	0.92 \pm 0.03	0.65 \pm 0.09	0.93 \pm 0.04
	Distractor	0.21 \pm 0.08	0.66 \pm 0.06	0.96 \pm 0.02	0.95 \pm 0.04	0.08 \pm 0.05	0.96 \pm 0.02	0.92 \pm 0.03	1.00 \pm 0.00
	mean	0.10 \pm 0.12	0.77 \pm 0.12	0.88 \pm 0.09	0.92 \pm 0.05	0.05 \pm 0.05	0.94 \pm 0.03	0.79 \pm 0.15	0.96 \pm 0.05
Can	In-domain	0.95 \pm 0.03	0.92 \pm 0.03	0.93 \pm 0.04	0.97 \pm 0.03	0.99 \pm 0.01	0.97 \pm 0.01	0.96 \pm 0.02	0.97 \pm 0.01
	Background	0.03 \pm 0.04	0.79 \pm 0.04	0.65 \pm 0.06	0.87 \pm 0.02	0.02 \pm 0.02	0.71 \pm 0.06	0.69 \pm 0.07	0.79 \pm 0.06
	Distractor	0.43 \pm 0.03	0.53 \pm 0.01	0.57 \pm 0.05	0.71 \pm 0.07	0.32 \pm 0.02	0.55 \pm 0.04	0.64 \pm 0.03	0.70 \pm 0.02
	mean	0.23 \pm 0.20	0.66 \pm 0.14	0.61 \pm 0.07	0.79 \pm 0.09	0.17 \pm 0.15	0.63 \pm 0.09	0.66 \pm 0.06	0.74 \pm 0.06
Square	In-domain	0.62 \pm 0.07	0.51 \pm 0.06	0.47 \pm 0.01	0.68 \pm 0.06	0.67 \pm 0.06	0.70 \pm 0.09	0.74 \pm 0.03	0.75 \pm 0.03
	Background	0.00 \pm 0.00	0.25 \pm 0.09	0.07 \pm 0.02	0.39 \pm 0.08	0.00 \pm 0.00	0.38 \pm 0.03	0.22 \pm 0.04	0.47 \pm 0.15
	Distractor	0.37 \pm 0.06	0.42 \pm 0.04	0.41 \pm 0.05	0.69 \pm 0.09	0.33 \pm 0.07	0.59 \pm 0.05	0.38 \pm 0.10	0.72 \pm 0.07
	mean	0.18 \pm 0.19	0.34 \pm 0.11	0.24 \pm 0.17	0.54 \pm 0.17	0.17 \pm 0.17	0.49 \pm 0.11	0.30 \pm 0.11	0.59 \pm 0.17
Transport	In-domain	0.49 \pm 0.01	0.37 \pm 0.05	0.41 \pm 0.01	0.46 \pm 0.07	0.61 \pm 0.01	0.59 \pm 0.05	0.65 \pm 0.10	0.58 \pm 0.02
	Background	0.00 \pm 0.00	0.05 \pm 0.02	0.01 \pm 0.01	0.05 \pm 0.02	0.00 \pm 0.00	0.14 \pm 0.04	0.05 \pm 0.02	0.20 \pm 0.06
	Distractor	0.21 \pm 0.03	0.21 \pm 0.03	0.23 \pm 0.02	0.34 \pm 0.11	0.24 \pm 0.08	0.35 \pm 0.03	0.39 \pm 0.05	0.41 \pm 0.02
	mean	0.11 \pm 0.11	0.13 \pm 0.08	0.12 \pm 0.11	0.20 \pm 0.17	0.12 \pm 0.13	0.25 \pm 0.11	0.22 \pm 0.17	0.30 \pm 0.11

Table 8: Performance of Random Overlay, SODA and RoboSaGA under visual domain shifts. Each is evaluated with BC-MLP, BC-RNN, across four simulated tasks against distractors and background variations.

Policy		Diffusion Policy			
Method		None	Overlay	SODA	SaGA
Task	Domain				
Lift	<i>In-domain</i>	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
	<i>Background</i>	0.86 ± 0.02	0.98 ± 0.02	0.98 ± 0.03	0.93 ± 0.03
	<i>Distractor</i>	0.55 ± 0.01	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
	<i>mean</i>	0.70 ± 0.16	0.99 ± 0.02	0.99 ± 0.02	0.97 ± 0.04
Can	<i>In-domain</i>	1.00 ± 0.00	0.98 ± 0.00	0.97 ± 0.02	0.97 ± 0.01
	<i>Background</i>	0.55 ± 0.04	0.87 ± 0.04	0.75 ± 0.02	0.91 ± 0.02
	<i>Distractor</i>	0.58 ± 0.00	0.75 ± 0.01	0.77 ± 0.01	0.86 ± 0.02
	<i>mean</i>	0.56 ± 0.03	0.81 ± 0.07	0.76 ± 0.02	0.89 ± 0.03
Square	<i>In-domain</i>	0.93 ± 0.01	0.91 ± 0.01	0.87 ± 0.01	0.91 ± 0.01
	<i>Background</i>	0.01 ± 0.01	0.59 ± 0.05	0.63 ± 0.07	0.76 ± 0.06
	<i>Distractor</i>	0.48 ± 0.07	0.89 ± 0.01	0.81 ± 0.05	0.90 ± 0.02
	<i>mean</i>	0.25 ± 0.24	0.74 ± 0.15	0.72 ± 0.11	0.83 ± 0.08
Transport	<i>In-domain</i>	0.91 ± 0.01	0.90 ± 0.03	0.90 ± 0.06	0.90 ± 0.02
	<i>Background</i>	0.00 ± 0.00	0.45 ± 0.03	0.33 ± 0.07	0.61 ± 0.05
	<i>Distractor</i>	0.44 ± 0.06	0.58 ± 0.04	0.61 ± 0.01	0.60 ± 0.06
	<i>mean</i>	0.22 ± 0.22	0.51 ± 0.08	0.47 ± 0.15	0.61 ± 0.06

Table 9: **Performance of Random Overlay, SODA and RoboSAGA under visual domain shifts.** Each is evaluated with Diffusion Policy, across four simulated tasks against distractors and background variations.

Appendix D: Interpretability Misalignment in Saliency Maps:

Here, we select three representative examples from the *Transport* task to illustrate the potential misalignment between input saliency from the policies and human interpretation.

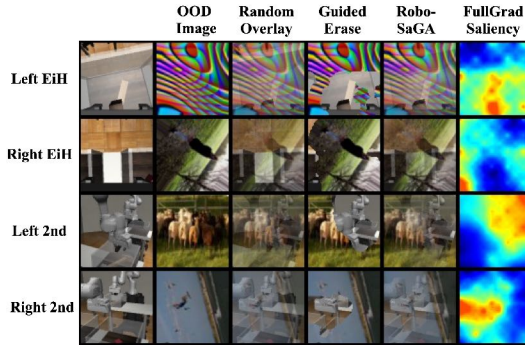


Figure 9: **Saliency Visualisation of *Transport* Task from BC-MLP.** *EiH*: eye-in-hand camera. *2nd*: second-person-camera

As observed in Fig. 9, the saliency maps produced by the history-independent BC-MLP align more closely with human intuition compared to those from BC-RNN (Fig. 10a) and the Diffusion Policy (Fig. 10b). In these visualisations, the Left Eye-in-Hand (EiH) camera focuses on the box handle, while the Right EiH camera does not provide task-critical information, exhibiting random focus instead. The left second-person camera focuses on the right arm, and the right second-person camera focuses on the right hand. Notably, the left second-person camera does not focus on the lid handle, as this piece of information is already provided by the left EiH camera. However, despite the alignment of the saliency maps with human intuition, experimental results indicate that this alignment does not necessarily translate into improved robustness against background variations. Indeed, none of the tested augmentation methods enhanced the performance of BC-MLP in the *Transport* task (see Tab. 8).

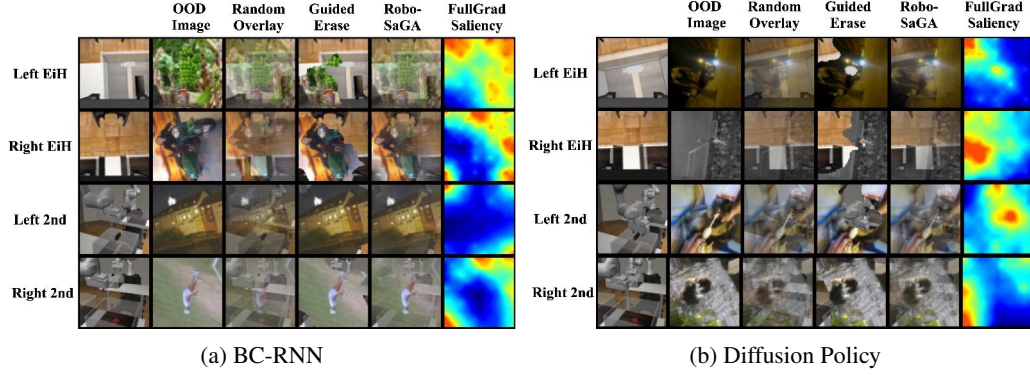


Figure 10: **Saliency Visualisation of *Transport* Task.** *EiH*: eye-in-hand camera. *2nd*: second-person-camera

459 Although achieving higher robustness against background variations, the saliency maps produced
 460 by the history-dependent BC-RNN and Diffusion Policy, in contrast to the history-independent BC-
 461 MLP, are less interpretable from a human perspective. These maps can focus on elements considered
 462 trivial by humans. Given that the task relies on historical observations, the robot’s states, multi-
 463 views, and exhibits varying levels of visual dependency throughout its execution, we argue that
 464 what humans perceive as visually important may not always align with the network’s focus.