Hybrid hierarchical learning for solving complex sequential tasks using the robotic manipulation network ROMAN

Citation for published version:
Triantafyllidis, E, Acero , F, Liu, Z & Li, A 2023, 'Hybrid hierarchical learning for solving complex sequential tasks using the robotic manipulation network ROMAN', *Nature Machine Intelligence*, vol. 5, no. 9, pp. 991-1005. https://doi.org/10.1038/s42256-023-00709-2

Digital Object Identifier (DOI):
10.1038/s42256-023-00709-2

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Publisher's PDF, also known as Version of record

Published In:
Nature Machine Intelligence

nature machine intelligence

Article

https://doi.org/10.1038/s42256-023-00709-2

Hybrid hierarchical learning for solving complex sequential tasks using the robotic manipulation network ROMAN

Received: 6 December 2022

Accepted: 18 July 2023

Published online: 7 September 2023

Check for updates

Eleftherios Triantafyllidis[1], Fernando Acero[2], Zhaocheng Liu[1] & Zhibin Li[1,2]

Solving long sequential tasks remains a non-trivial challenge in the field of embodied artificial intelligence. Enabling a robotic system to perform diverse sequential tasks with a broad range of manipulation skills is a notable open problem and continues to be an active area of research. In this work, we present a hybrid hierarchical learning framework, the robotic manipulation network ROMAN, to address the challenge of solving multiple complex tasks over long time horizons in robotic manipulation. By integrating behavioural cloning, imitation learning and reinforcement learning, ROMAN achieves task versatility and robust failure recovery. It consists of a central manipulation network that coordinates an ensemble of various neural networks, each specializing in different recombinable subtasks to generate their correct in-sequence actions, to solve complex long-horizon manipulation tasks. Our experiments show that, by orchestrating and activating these specialized manipulation experts, ROMAN generates correct sequential activations accomplishing long sequences of sophisticated manipulation tasks and achieving adaptive behaviours beyond demonstrations, while exhibiting robustness to various sensory noises. These results highlight the significance and versatility of ROMAN's dynamic adaptability featuring autonomous failure recovery capabilities, and underline its potential for various autonomous manipulation tasks that require adaptive motor skills.

When humans interact with their surrounding environment, they perform highly complex in-sequence tasks with seemingly minimal effort[1–3]. By virtue of our highly complex cognition, solving complex sequences of manipulation tasks appears to require very little effort[4,5].

In contrast, observing the above from the perspective of robots as agents with embodied intelligence, achieving these physical interactions is currently far from trivial[5,6] and solving complex sequential tasks over a long horizon remains an ongoing challenge[7,8]. Notably, a task as simple as retrieving a glass from a shelf, pouring in water and placing it onto a table may seem trivial, but from an embodied intelligence perspective remains considerably challenging. Essentially, successful sequential manipulation is achieved when (1) high-level skills are satisfied, (2) sensory events are predicted, (3) the end goals are known and (4) the sequences of different skills are conceptualized in our minds and more broadly by our nervous system[3,9].

Nevertheless, robots can perform repetitive manipulation tasks with high precision, provided that these are confined to specific tasks[10,11]. Some of these tasks include picking and placing[4,12], swing peg

[1]School of Informatics, University of Edinburgh, Edinburgh, UK. [2]Department of Computer Science, University College London, London, UK. e-mail: alex.li@ucl.ac.uk

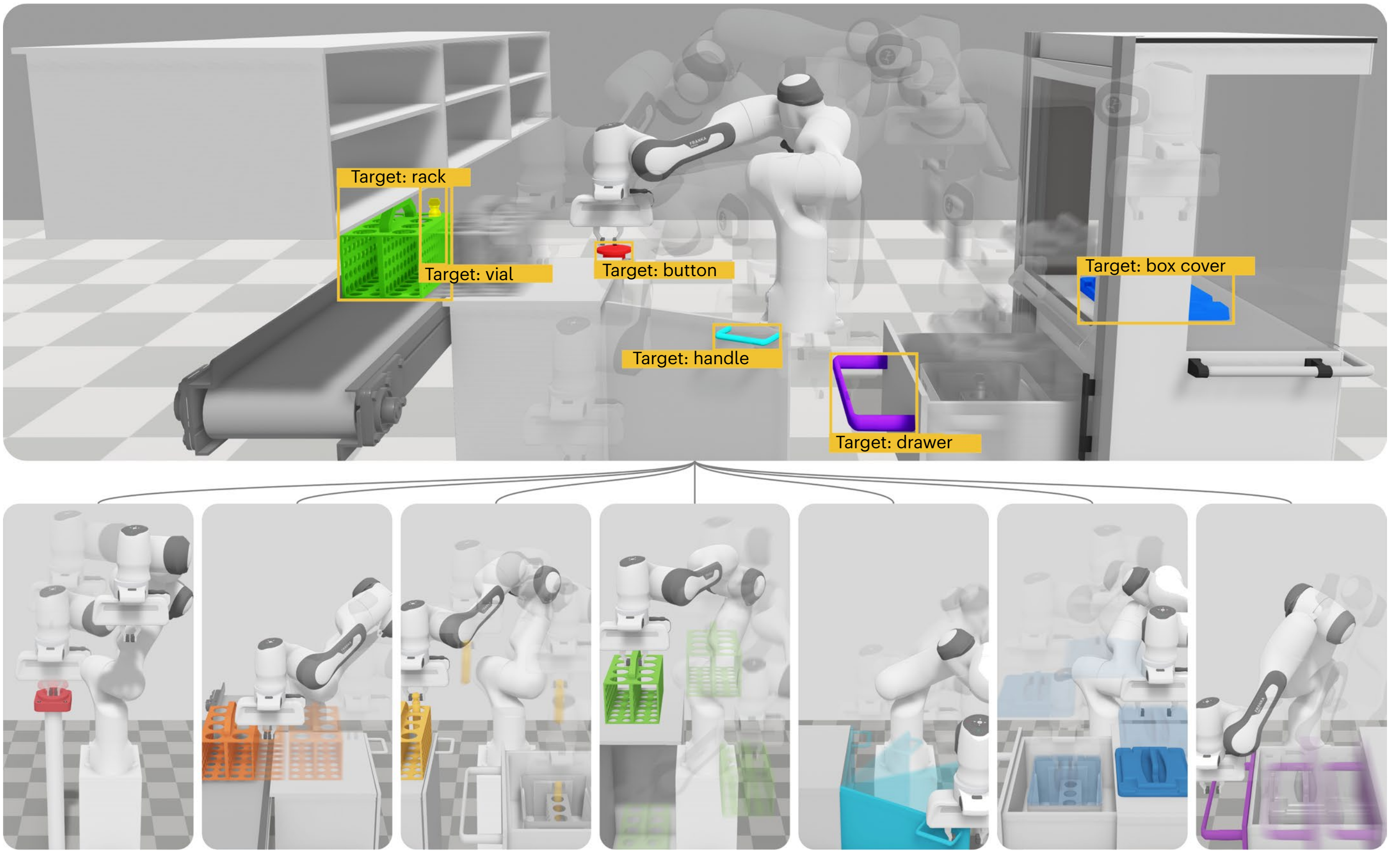

Fig. 1 | Capabilities of the ROMAN framework. An HHL framework for hierarchical task learning, capable of solving long-time-horizon tasks that require successful activation and coordination of diverse expert skills to solve a sequence of non-interrelated tasks commonly necessary in robotics and physical interactions. The derivation of high-level specialized experts in ROMAN allowed us to construct a gating network that is trained for elevated task-level scene understandings, for the planning of complex sequential long-time-horizon tasks and for the successful and timely activation of low-level expert networks. We studied a set of seven specialized manipulation skills that are common in daily life and can be combined to create a higher level of manipulation skills. These specialized skills included (1) pushing a button, (2) pushing, (3) picking and inserting, (4) picking and placing, (5) rotating–opening, (6) picking and dropping and (7) pulling–opening. Unlike conventional planning methods or state machines, ROMAN exhibits adaptability in (1) randomized task sequences, (2) generalization outside demonstrated cases and (3) recovery and robustness against local minima. The ability of the gating network to achieve such versatility is attributed to (1) the HHL architecture in ROMAN's core framework and (2) the high-level task decomposition of complex sequences by the various experts in the framework, allowing the central MN, which is a gating network, to be trained on high-level scene understanding and orchestrations of experts. The system architecture is based on an MoE-based architecture, which is able to successfully adapt to environmental demands, overcome various levels of uncertainties and most importantly learn with minimal human imitation.

in hole[13,14], catching in-flight objects[15], insertion[14,16] or solving a Rubik's cube[17]. However, when it comes to solving a sequence of multiple tasks that vary in complexity, substantial challenges arise[11].

To overcome these limitations, we developed the novel robotic manipulation network ROMAN, which is an event-based hybrid hierarchical learning (HHL) framework, visualized in Fig. 1, for hierarchical task learning. This mixture of experts (MoE)-based hierarchical approach is capable of solving complex long-horizon manipulation tasks. We evaluated the framework in simulation and validated its robustness during long-horizon sequential tasks against sensory uncertainties. Thereafter, we performed extensive ablation studies of the internal learning procedure, evaluated the effects of different demonstrations and benchmarked the performance of ROMAN when compared with monolithic neural networks (NNs). Our results demonstrate that, by recombining and fusing ROMAN's core experts and skills together, our framework is able to solve considerably complex, long-horizon sequential manipulation tasks, commonly encountered in our everyday lives, with generalizing capabilities. In the remainder of this Article, we review the related work, present ROMAN's results, discuss future work and elaborate on the technical details of our methodology.

Real-world impact of intelligent robotics

Pre-programming robots via analytical models can lead to suboptimal solutions due to simplified modelling of real-world dynamics, and online recomputation can be expensive and unable to account for dynamically changing physical properties. Current advances in artificial intelligence and machine learning offer a promising avenue to advance robot learning and embodied intelligence[12,14,18,19].

The common reinforcement learning (RL) algorithms among related work are Proximal Policy Optimization (PPO)[20] and Soft Actor–Critic[21]. Although PPO is on policy and generally less sample efficient than off-policy algorithms such as Soft Actor–Critic, PPO is less prone to instabilities and typically requires less hyperparameter tuning than Soft Actor–Critic[20–22]. For these reasons, we chose PPO as our RL algorithm.

Imitation learning and learning from demonstration

RL algorithms face challenges in dealing with complex tasks, particularly when rewards are sparse, which exacerbates the exploration–exploitation trade-off[23–25]. One major limitation is the need to generate their own experience from scratch[26,27], which can require millions of state transitions and days of training due to the absence of prior knowledge[19,28].

An alternative is to use imitation learning (IL), inspired by the prior knowledge that humans possess when learning motor tasks instead of starting from scratch[29], whereby agents learn to emulate the demonstrated behaviour. This is also known as learning from demonstration, showing promising results in dexterous robotic tasks that would have been impossible to pre-program or substantially difficult to learn via conventional RL, due to the required degree of exploration and the necessity to carefully craft rewards for the desired behaviour[12,23,26].

Most IL and learning from demonstration approaches depend on demonstrations from human experts. While some forms of demonstration could be substituted via conventional trajectory optimization[12,30] or RL[31–33], these methods generally require carefully designed costs or rewards and considerable interaction time between the robot and the environment.

One of the main IL algorithms used in related work is Behavioural Cloning (BC), which performs supervised learning on the policy from a set of demonstrated state–action transitions, showing promising success in robotic tasks[8,12,34,35]. However, BC has numerous limitations when used in isolation, such as lack of exploration, limited robustness towards new non-encountered states and dependence on large, near-optimal demonstrations[36].

Naively copying expert demonstrations via BC is prone to problematic performance when the agent visits states not encountered in the demonstrations due to covariate shifting errors that compound over time, which drives the need for large numbers of demonstrations[36,37], leading to operator fatigue and hence degraded performance[4,38]. Even from a biological perspective, the sole and naive dependence on an expert to learn new skills is misguided[25,27,39]. Zaadnoordijk et al. provided a matching analogy whereby trial and error is a crucial part of our early lives: "Human infants are in many ways a close counterpart to a computational system learning in an unsupervised manner, as infants too must learn useful representations from unlabeled data"[25]. For machine learning, this suggests that learning in its core should not entirely depend on copying an 'expert', but rather encourage further exploration beyond imitation, to draw inspiration from a neurobiological standpoint[27,39].

An alternative to overcome some of the limitations of BC is inverse RL, which infers the underlying reward function in observed demonstrations to explain the demonstrations and achieve a near-optimal behaviour[36,40,41]. One of the popular inverse RL algorithms is Generative Adversarial Imitation Learning (GAIL)[36]. In this framework, GAIL uses a second NN, known as a discriminator, responsible for distinguishing between agent- and expert-generated trajectories[36].

Hierarchical learning

Solving complex tasks using monolithic NNs through RL or IL can be challenging due to (1) long-horizon problems, whereby the computational complexity of approximating a policy is high, (2) the variability of the task requiring numerous subtasks and (3) sample complexities of dexterous tasks[7,8,42–44]. Moreover, the successful completion of a long-time-horizon task is contingent upon the successful completion of all subtasks in a particular sequence[44]. Finally, even using smaller subtasks to solve the problem[44,45] can still be aggravated by considerable variations in their nature and limited task interrelation[46].

Hierarchical learning (HL), whether used for RL or IL, can mitigate the above problems and alleviate some of these complexities[19,47–49]. HL offers multiple benefits when it comes to complex tasks associated with sparse rewards[7], as it allows the decomposition of tasks into more approachable problems, that is, subtasks[8]. When these HL policies implement IL, commonly referred to as HIL, the differentiation between the specialized experts and the acquisition of specialized human skills in a teacher–student fashion is considered easier[8,42,50].

A popular approach is the use of MoEs, where multiple task-specific experts are trained and specialized on a given subtask, with applications in computer graphics[18,51] and robotics[8,19,52]. However, hierarchical reinforcement learning (HRL) still fundamentally depends on RL and hence is adversely affected by sparse rewards, complex planning tasks and difficulty in using prior knowledge[8,44]. HIL[8,42] leverages expert demonstrations, unlike RL or HRL, to aid the overall training process and allow the demonstrator to isolate subtasks to facilitate solving longer, more complex and in-sequence tasks[8,50].

Currently, in robotic manipulation, methods using MoEs trained with HRL or HIL are limited in the state of the art[44,45]. On the basis of previous work that introduced ensemble techniques in robot locomotion[19] and human-centred teleoperation[38], we are motivated to explore a new approach of IL using human-demonstrated tasks developing a suitable MoE architecture in the domain of robotic manipulation. This approach has the potential to extend beyond the original demonstrations and enable more complex manipulation tasks. Work similar to ours used an HRL approach to train a robotic gripper incorporating three experts: (1) approach, (2) manipulate and (3) retract[44]. While their results were validated against BC, showing higher (90%+) success rates when compared with RL, these studied tasks were limited to non-sequential tasks with short time horizons on a manipulator with a lower number of degrees of freedom, and restricted to three experts solving only picking and placing tasks[44]. In contrast, our work can train a single expert capable of solving picking and placing, and when combined with other experts specialized in rather high-level subtasks when compared with ref. 44 we can solve complex and long-horizon sequential tasks in manipulation.

Results

This section presents the results for the ROMAN framework, which is composed of a modular hybrid hierarchical architecture to combine adaptive motor skills for solving complex manipulation tasks. It features a central manipulation network (MN) that activates specialized task-level experts in a required sequential combination, resulting in higher levels of manipulation capability and improved generalization to non-demonstrated situations. Moreover, the MN exhibits recovery capabilities by activating multiple expert weights to overcome local minima, which ultimately enhances the robustness for solving long-horizon sequential tasks.

Specifically, our validation shows the robustness of ROMAN's HHL approach against (1) high exteroceptive observation noise, (2) complex non-interrelated compositional subtasks, (3) long-time-horizon sequential tasks and (4) cases not encountered during the demonstrated sequences. ROMAN achieves behaviour beyond imitation through hybrid training and allows the dynamic coordination of experts to recover from local minima successfully, with examples depicted in Fig. 2. Our findings highlight the versatility and adaptability of ROMAN, enabling autonomous manipulation with adaptive motor skills.

To evaluate the scalability of a hierarchical architecture versus a single-NN approach, we compared ROMAN's preliminary two-dimensional (2D) and final three-dimensional (3D) hierarchical architecture stage against monolithic NNs sharing an equivalent hybrid learning procedure. Snapshots of ROMAN completing long-horizon sequential tasks can be seen in Fig. 3, with examples of 2D and 3D operation depicted in Fig. 3c,d, respectively. Thereafter, we evaluated ROMAN's final 3D stage composed of seven experts against (1) different levels of exteroceptive uncertainty, (2) extensive ablation studies of the internal hybrid learning procedure and (3) the effects of different numbers of demonstrations provided to the framework. All subsequent results from the experiments were conducted with identical network settings (states, actions and rewards), number of demonstrations and hyperparameter values to retain consistency. The overall architecture of ROMAN is depicted in Fig. 4, with the state space and settings of

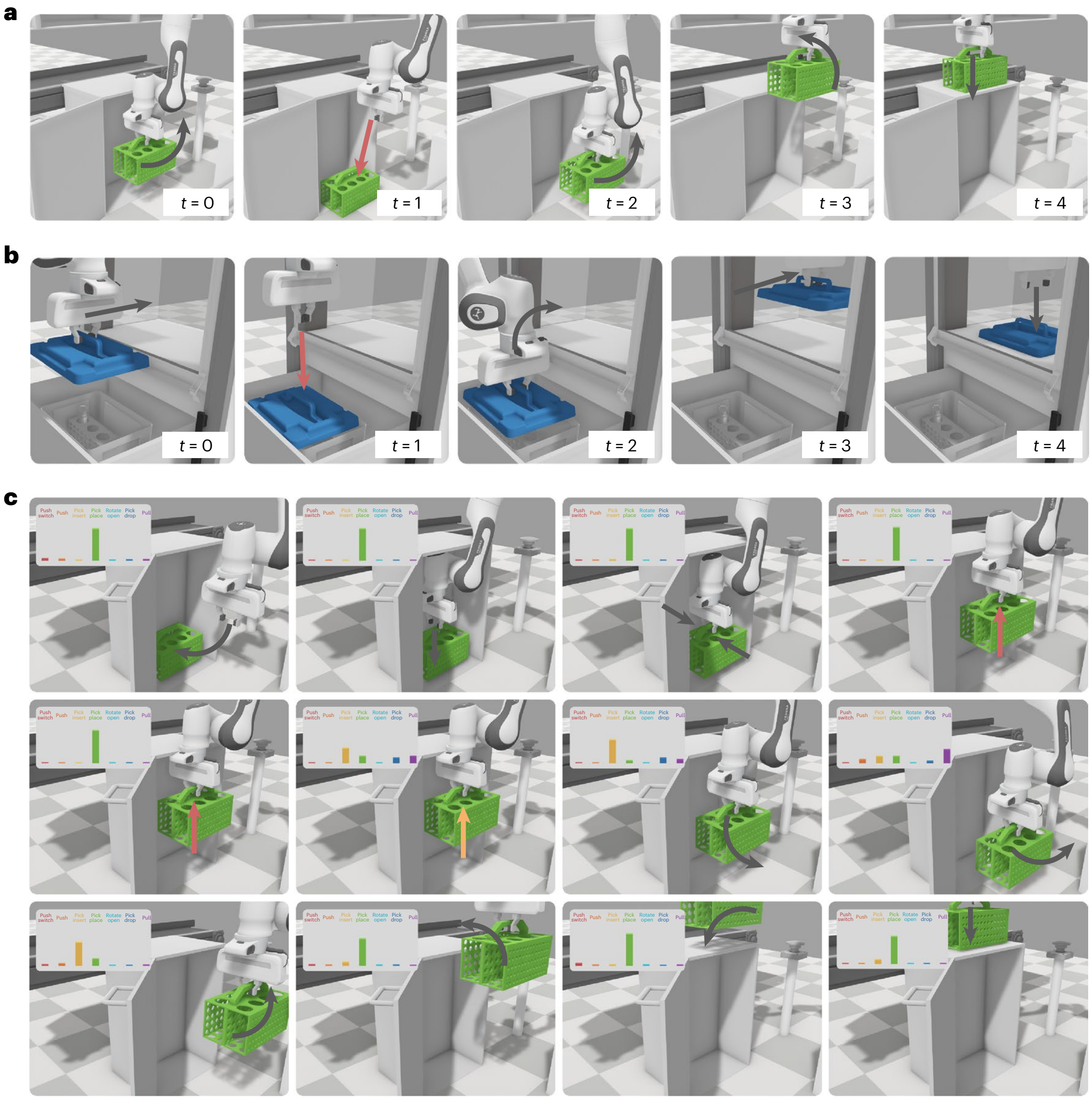

Fig. 2 | ROMAN demonstrated the ability to adapt to the scenarios beyond demonstrated sequences and exhibited dynamic recovery capabilities, by balancing exploitation and exploration via the HHL approach. a,**b**, Policy adaptation of ROMAN during failures in pick and place and pick and drop subtasks, respectively. These intermediate failures are attributed to either an expert or a gating network error. In these instances, we show infrequent error cases (*t* = 1) of these experts, which, however, quickly re-adapt and regrasp the items (*t* = 2 to *t* = 4) to successfully complete the sequence. Most notably, this can be due to a combination of incorrect grasping of objects, expert trajectories or activation of sequences. **c**, The ability of the MN of the ROMAN framework to dynamically adapt in cases that were not encountered in the initial demonstrations, but rather those states were visited during RL training as the balance of exploitation and exploration, ultimately exhibiting new behaviours beyond imitation, leading to recovery capabilities from local minima. The figure represents 12 snapshots in time with a sequence from left to right and top to bottom, and the weight assignments by the MN highlighted.

each NN specified in Table 1. More details on the hyperparameters and dimensions of the networks can be found in Supplementary Information and more specifically in Supplementary Tables 12, 13, 14 and 15. Information on the demonstrations can be found in Methods.

Definition of success rate

Task success was attained and defined when all seven subtask goals depicted in Fig. 1 were satisfied. Consequently, to consider a scenario successful, all interrelated subtasks needed to be sequentially completed within the time limit.

Limitations of monolithic networks in long-horizon tasks

ROMAN's preliminary version in two dimensions consisted of five experts (Fig. 3c) and was thereafter scaled up to three dimensions consisting of seven experts (Figs. 1 and 3d). Consequently, in this section we compare ROMAN's preliminary and final stages against two monolithic single NNs with an equivalent hybrid learning procedure (shown in Fig. 4a) for two and three dimensions, respectively. These baseline evaluations allowed a direct comparison of a monolithic versus a hierarchical approach, to evaluate and demonstrate the advantages of a hierarchical task decomposition with an identical learning procedure. The single NNs had states identical to those of ROMAN's MN and actions identical to those of ROMAN's experts. To conduct a fair comparison, a total of *N* = 100 and *N* = 140 demonstrations were provided to the single NNs, accounting for ROMAN's 2D and 3D cases composed of five and seven experts pretrained with *N* = 20 demonstrations, respectively. Table 2 details the robustness of ROMAN's 3D case, including individual expert success rates.

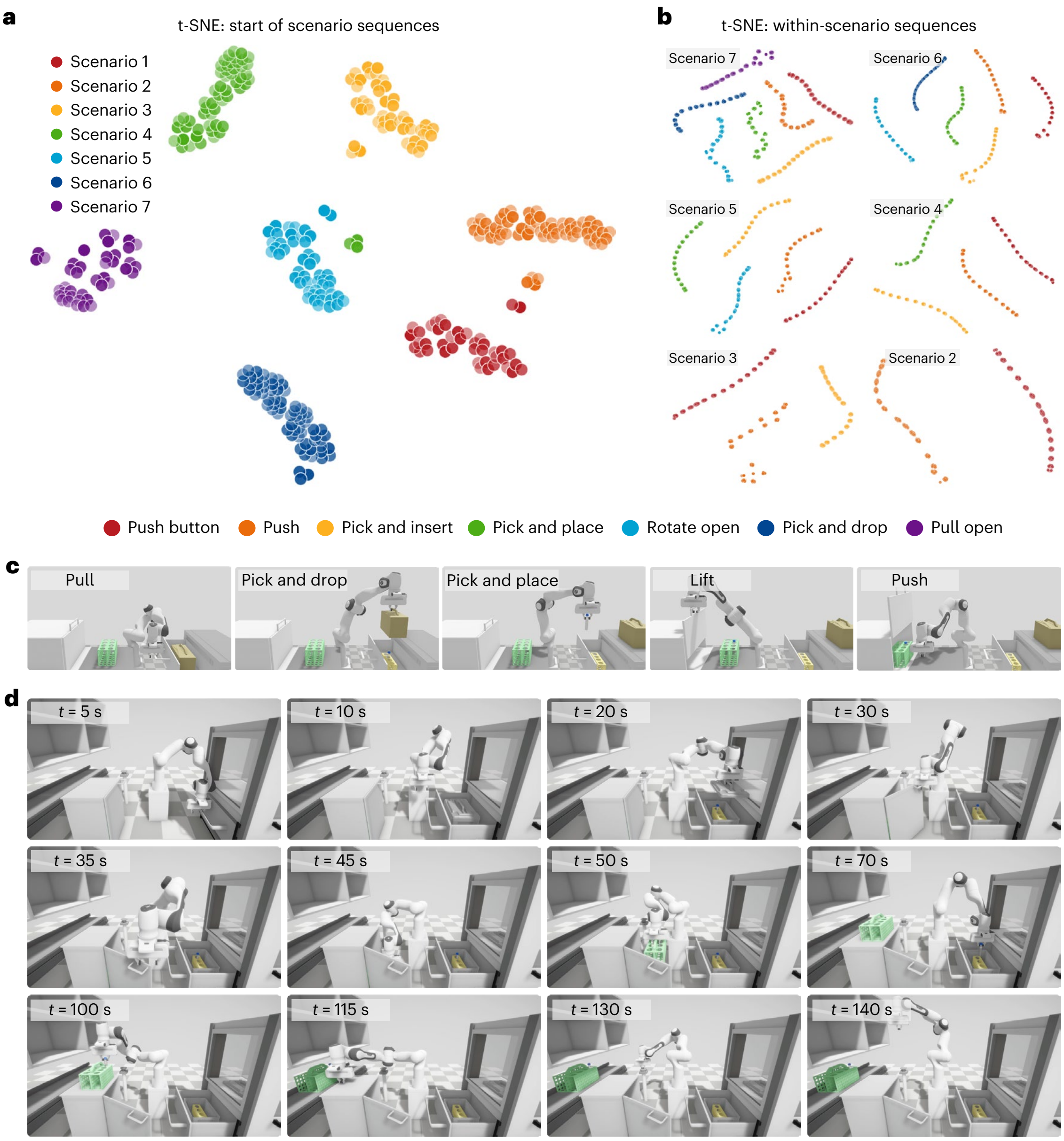

Fig. 3 | Analysis of the MN observations using the t-SNE, with visualized snapshots showing ROMAN's completion of sequential tasks in 2D and 3D scenarios. The t-SNE projects the 29-dimensional MN state vector into two dimensions. Principal component analysis was used to warm-start the t-SNE projection. **a**, The depiction of the state vectors at the start of each of the seven case scenarios, sampled at 1,000 Hz for 1 s. A total of 1,000 samples were projected with a perplexity of 400. **b**, An illustration of the state vectors during the sequence of actions contained in each case scenario, sampled for the first 1.5 s of each expert sequence. Consequently, as these are sampled within the sequence of actions, they appear 'trajectory'-like, since the robot and the objects manipulated by it are in motion during the sampling. A total of 1,500 samples were projected with a perplexity of 200. Six out of seven scenario cases are depicted, as in practice S1 only includes a single expert activation and hence was omitted from the analysis. **c**, ROMAN in its initial 2D stage depicting all five distinct subtasks managed by each expert. **d**, ROMAN in its final stage in the most complex setting and longest-time-horizon sequential tasks in full 3D space, with seven different experts.

The results shown in Table 3 and Table 4, for the 2D and 3D cases of the monolithic NN, respectively, suggest that a single NN is unable to solve the complex nature and long sequential task of our validated manipulation scenario given the same training procedure. While in two dimensions the single NN attains high success rates to some extent, these remain substantially lower than ROMAN's, especially in increasing time horizons (S3, S4 and S5). Extending the dimensionality to three dimensions reveals that a monolithic NN is mostly unable to attain robust performance (S3), exhibiting complete failure in longer and more complex sequential cases (S4 and beyond). These results highlight the value of a hierarchical task decomposition as with ROMAN's architecture. For more details and expansion regarding the monolithic NNs, including their architecture and hyperparameters, please consult Supplementary Tables 14 and 15.

Validation against exteroceptive uncertainty

This section presents all the results tested in 3D space with seven experts, with details shown in Table 1, to study the domain of robotics with complex settings. While scaling up to three dimensions with seven experts, the first objective was to evaluate the robustness of the hierarchical framework against different levels of Gaussian-distributed exteroceptive noise on the position states. The rationale for introducing

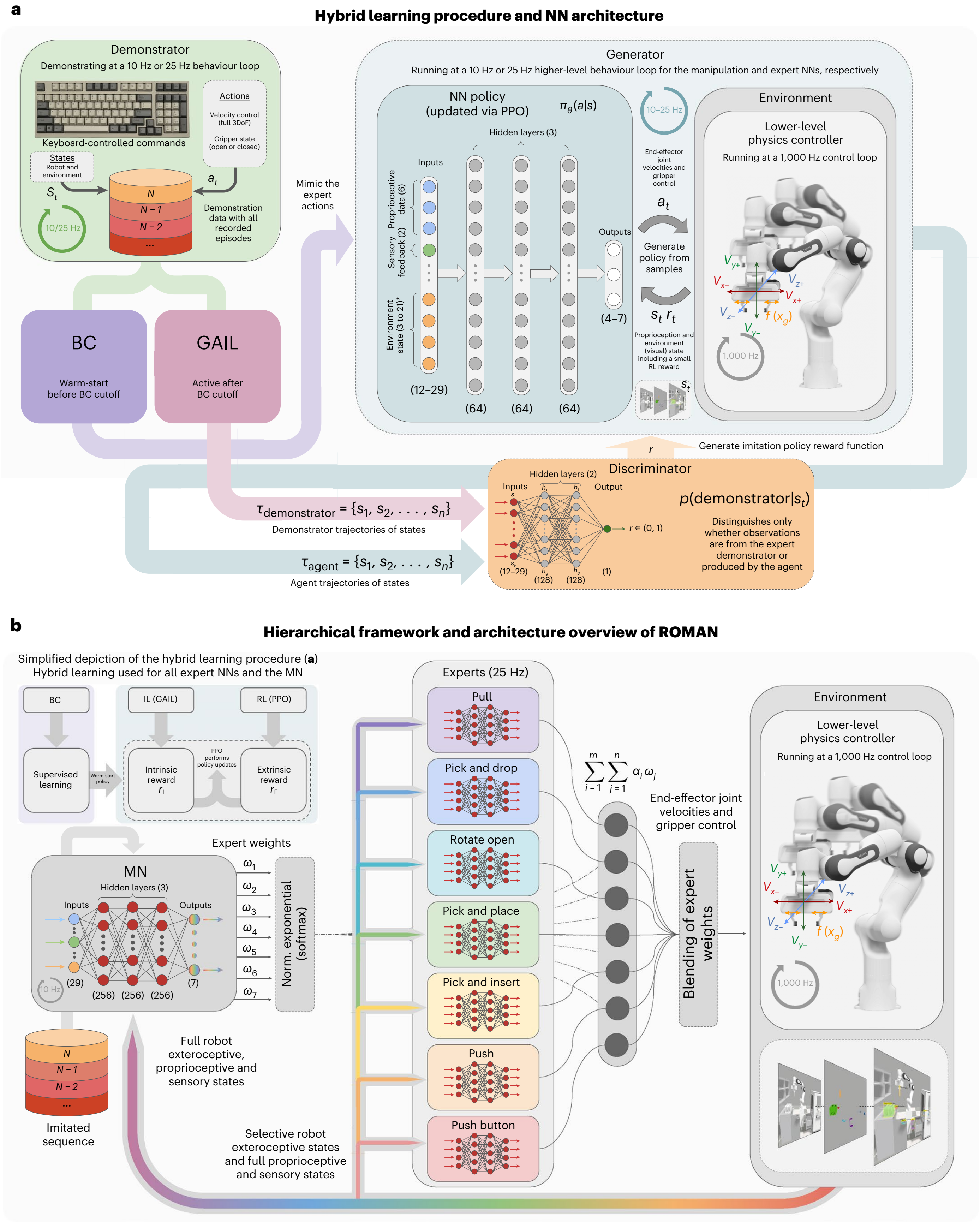
a
Hybrid learning procedure and NN architecture
Demonstrator
Demonstrating at a 10 Hz or 25 Hz behaviour loop
Keyboard-controlled commands
Actions
Velocity control (full 3DoF)
Gripper state (open or closed)
States
Robot and environment
s_t
a_t
N
N – 1
N – 2
...
10/25 Hz
Demonstration data with all recorded episodes
BC
Warm-start before BC cutoff
GAIL
Active after BC cutoff
Mimic the expert actions
Generator
Running at a 10 Hz or 25 Hz higher-level behaviour loop for the manipulation and expert NNs, respectively
NN policy (updated via PPO)
$\pi_\theta(a|s)$
Hidden layers (3)
Inputs
Proprioceptive data (6)
Sensory feedback (2)
Environment state (3 to 21)*
Outputs
(12–29)
(64)
(64)
(64)
(4–7)
10–25 Hz
End-effector joint velocities and gripper control
a_t
Generate policy from samples
s_t r_t
Proprioception and environment (visual) state including a small RL reward
s_t
Environment
Lower-level physics controller
Running at a 1,000 Hz control loop
V_{y+}
V_{x-}
V_{z+}
V_{x+}
V_{z-}
V_{y-}
$f(x_g)$
1,000 Hz
Generate imitation policy reward function
r
Discriminator
Hidden layers (2)
Inputs
Output
s_1
s_t
h_1
h_g
(12–29)
(128)
(128)
(1)
$r \in (0, 1)$
$p(\text{demonstrator}|s_t)$
Distinguishes only whether observations are from the expert demonstrator or produced by the agent
$\tau_{\text{demonstrator}} = \{s_1, s_2, \ldots, s_n\}$
Demonstrator trajectories of states
$\tau_{\text{agent}} = \{s_1, s_2, \ldots, s_n\}$
Agent trajectories of states
b
Hierarchical framework and architecture overview of ROMAN
Simplified depiction of the hybrid learning procedure (a)
Hybrid learning used for all expert NNs and the MN
BC
IL (GAIL)
RL (PPO)
Supervised learning
Warm-start policy
Intrinsic reward r_I
PPO performs policy updates
Extrinsic reward r_E
Expert weights
MN
Hidden layers (3)
Inputs
Outputs
(29)
(256)
(256)
(256)
(7)
10 Hz
ω_1
ω_2
ω_3
ω_4
ω_5
ω_6
ω_7
Norm. exponential (softmax)
N
N – 1
N – 2
...
Imitated sequence
Full robot exteroceptive, proprioceptive and sensory states
Selective robot exteroceptive states and full proprioceptive and sensory states
Experts (25 Hz)
Pull
Pick and drop
Rotate open
Pick and place
Pick and insert
Push
Push button
$\sum_{i=1}^{m}\sum_{j=1}^{n} \alpha_i \omega_j$
End-effector joint velocities and gripper control
Blending of expert weights
Environment
Lower-level physics controller
Running at a 1,000 Hz control loop
V_{y+}
V_{x-}
V_{z+}
V_{x+}
V_{z-}
V_{y-}
$f(x_g)$
1,000 Hz

Fig. 4 | Hybrid hierarchical architecture of ROMAN composed of high-level experts and the gating NN, and the formation of the ROMAN framework. **a**, The hybrid learning architecture of each high-level expert and gating NN. DoF, degrees of freedom. **b**, The higher hierarchical formation of ROMAN and how the experts are orchestrated and activated. The multilayer perceptrons for the NNs are visually depicted in both panels.

Table 1 | Summary of ROMAN's overall NN architecture, the state space of each NN and the settings of individual components in the hierarchical framework

Network architecture, characteristics and demonstration settings							
Master (MN)	**Expert NNs**						
	Push button	**Push**	**Pick and insert**	**Pick and place**	**Rotate open**	**Pick and drop**	**Pull open**
State space (vector size)							
Total: 29	**Total: 11**	**Total: 14**	**Total: 14**	**Total: 14**	**Total: 11**	**Total: 14**	**Total: 11**
Agent position (3)	Agent position (3)	Agent position (3)	Agent position (3)	Agent position (3)	Agent position (3)	Agent position (3)	Agent position (3)
Agent velocity (3)	Agent velocity (3)	Agent velocity (3)	Agent velocity (3)	Agent velocity (3)	Agent velocity (3)	Agent velocity (3)	Agent velocity (3)
Gripper force (2)	Gripper force (2)	Gripper force (2)	Gripper force (2)	Gripper force (2)	Gripper force (2)	Gripper force (2)	Gripper force (2)
Full environment (21)	Button position (3)	Rack position (3)	Rack position (3)	Rack position (3)	Cabinet position (3)	Box position (3)	Drawer position (3)
		Conveyor position (3)	Vial position (3)	Rack target (3)		Unbox target (3)	
Action space (vector size)							
Total: 7	**Total: 4**	**Total: 4**	**Total: 4**	**Total: 4**	**Total: 4**	**Total: 4**	**Total: 4**
Agent weights (7)	Agent velocity (3)	Agent velocity (3)	Agent velocity (3)	Agent velocity (3)	Agent velocity (3)	Agent velocity (3)	Agent velocity (3)
	Gripper state (1)	Gripper state (1)	Gripper state (1)	Gripper state (1)	Gripper state (1)	Gripper state (1)	Gripper state (1)
Demonstration settings and training times (number, demo time, train time)							
N=42 (N=6 per case)	N=20	N=20	N=20	N=20	N=20	N=20	N=20
$t_{\text{demo}}\approx$42 min	$t_{\text{demo}}\approx$7 min	$t_{\text{demo}}\approx$6 min	$t_{\text{demo}}\approx$12 min	$t_{\text{demo}}\approx$10 min	$t_{\text{demo}}\approx$7 min	$t_{\text{demo}}\approx$9 min	$t_{\text{demo}}\approx$7 min
t_{train}=11 h 22 min	t_{train}=3 h 1 min	t_{train}=3 h 59 min	t_{train}=23 h 30 min	t_{train}=11 h 46 min	t_{train}=2 h 39 min	t_{train}=3 h 43 min	t_{train}=3 h 18 min

Overview of the state and action spaces, as well as the demonstrations provided for each NN in ROMAN. A total of N=20 demonstrations were provided to pretrain the expert NNs in ROMAN, and a total of N=42 demonstrations to the MN, corresponding to N=6 demonstrations for each of the seven sequential cases.

noise in the exteroceptive states was to thoroughly evaluate the robustness of the framework against uncertainties under realistic conditions, since such states are typically more prone to noise than proprioceptive states in robotic systems[19].

Evaluation against increasing levels of Gaussian noise. Foremost, we validate each expert's individual robustness, which is critical before evaluating the MN's performance during sequential activation, to avoid failures being caused by individual expert performance. This minimized the covariance between the success rates of each expert and that of the MN. Table 2 shows that all individual experts, even when presented with higher levels of noise, are resilient against the tested levels of uncertainty. It is worth noting that all picking experts were slightly more prone to errors due to their higher complexity, in line with refs. 44,53.

Next, we evaluated the MN's performance in coordinating the different experts in the hierarchy of ROMAN. From the given seven experts, we tested seven different randomized case scenarios, where each scenario requires addition of another expert, making the overall tasks more complex. Results in Table 2 show robust performance to different noise levels. Although adding more experts increases the dimensionality of the problem, our results show that the MN is sufficiently resilient in the most complex settings in scenarios 6 and 7. However, there still is a performance drop in scenarios 3, 4 and 5 when compared with 6 and 7, which is discussed in Results.

Evaluation of vision system. The next objective was to test the robustness of ROMAN against exteroceptive uncertainties from a simulated vision system in the simulation. ROMAN and its experts, including the MN, were not trained with this vision detection module, but rather directly evaluated on it to test the feasibility and robustness of the framework to such a vision-based detection system. More details of the vision system can be found in Methods.

The results in Table 2 show that using a pretrained object detection module from vision produces high success rates even amongst the most complex sequential tasks. Despite a slight decrease in success rates as more sequences are added, ROMAN exhibits robustness to the vision system, sustaining high success levels. The decrease in success rates in S6 and less in S7 can be attributed to the unboxing subtask, which is more prone to visual occlusion (Fig. 1) and the similarity in the exteroceptive observations later analysed in a *t*-distributed stochastic neighbour embedding (t-SNE; Fig. 3).

Ablation study on ROMAN's default learning approach

The next validation entails a comparison with state-of-the-art learning paradigms, including HRL and HIL approaches, similar to related work[12,44]. ROMAN makes use of BC to warm-start the policy via supervised learning and thereafter uses intrinsic r_{I} (IL: GAIL) and extrinsic r_{E} (RL) rewards via PPO for training, and we conduct ablations to the training procedure by excluding at least one of the previous paradigms.

The ablation results in Table 5 show that the exclusive use of r_{E} (RL) exhibited complete failure, suggesting that the high complexity of the tasks is unattainable via random exploration of the action space. Using the r_{I} provided by GAIL or coupling it with r_{E} for RL and GAIL both showed substantially higher success rates, but limited to S1–S3, with longer-horizon tasks still being unattainable.

From the related work[7,12,44], we summarize that training with BC alone appears to yield rapid performance degradation as the time horizon increases. This is in line with our results for both BC and RL,BC at $\sigma=\pm0.5$ cm noise. While a notable boost in success rates is observed, longer sequential tasks such as S4–S7 (which exhibit higher variance in the trajectories visited due to compounding of errors throughout

Table 2 | Summary of the results evaluated on increasing levels of Gaussian noise and uncertainties from the vision system for each expert and the main MN in ROMAN

Individual expert success rates (10,000 trials per cell)							
Success (%)	**Push button**	**Push**	**Pick and insert**	**Pick and place**	**Rotate open**	**Pick and drop**	**Pull open**
$\sigma = \pm 0.0$ [cm]	0.996	0.998	0.919	0.986	0.999	0.997	0.970
$\sigma = \pm 0.5$ [cm]	0.999	0.999	0.933	0.989	0.999	0.994	0.993
$\sigma = \pm 1.0$ [cm]	0.999	0.999	0.939	0.982	0.999	0.994	0.985
$\sigma = \pm 1.5$ [cm]	1.000	0.999	0.920	0.965	0.999	0.988	0.969
$\sigma = \pm 2.0$ [cm]	0.999	0.998	0.872	0.941	0.999	0.973	0.962
$\sigma = \pm 2.5$ [cm]	0.999	0.991	0.826	0.903	0.998	0.955	0.950
MN success rates (1,000 trials per cell)							
Success (%)	**Scenario 1**	**Scenario 2**	**Scenario 3**	**Scenario 4**	**Scenario 5**	**Scenario 6**	**Scenario 7**
	One expert	**Two experts**	**Three experts**	**Four experts**	**Five experts**	**Six experts**	**Seven experts**
	(Push button)	**(+ Push)**	**(+ Pick and insert)**	**(+ Pick and place)**	**(+ Rotate open)**	**(+ Pick and drop)**	**(+ Pull open)**
$\sigma = \pm 0.0$ [cm]	0.976	0.972	0.847	0.951	0.728	0.954	0.903
$\sigma = \pm 0.5$ [cm]	0.973	0.975	0.817	0.959	0.794	0.960	0.952
$\sigma = \pm 1.0$ [cm]	0.977	0.990	0.798	0.946	0.776	0.933	0.939
$\sigma = \pm 1.5$ [cm]	0.980	0.986	0.720	0.846	0.722	0.836	0.841
$\sigma = \pm 2.0$ [cm]	0.967	0.986	0.737	0.837	0.753	0.820	0.815
$\sigma = \pm 2.5$ [cm]	0.973	0.986	0.723	0.763	0.697	0.719	0.744
MN vision system success rates (100 trials per cell)							
Success (%)	**Scenario 1**	**Scenario 2**	**Scenario 3**	**Scenario 4**	**Scenario 5**	**Scenario 6**	**Scenario 7**
Visual	0.97	0.96	0.82	0.83	0.79	0.51	0.72

The success rates for all individual experts and the MN in ROMAN for the 3D setting, across all scenarios, based on increased levels of Gaussian noise in the exteroceptive position observations. Additionally, we further tested the feasibility and robustness of the trained models by evaluating their performance directly on a vision system that provides exteroceptive information.

the trajectory) show lower performance when compared with that of ROMAN's default learning. Despite BC being a simple yet effective algorithm, its performance is greatly affected when presented with out-of-distribution states, in line with refs. 36,45,54. To further test this finding, we evaluate both BC and RL,BC on increased levels of noise of $\sigma = \pm 1.0$ cm and $\sigma = \pm 2.0$ cm.

At a noise level of $\sigma = \pm 1.0$ cm, we observe a slight drop in success rates for both BC and RL,BC. ROMAN's default settings still attain the highest success rates. At a higher level of $\sigma = \pm 2.0$ cm noise, we observe a notable drop in success rates for both BC and RL,BC. Employing BC at such levels of uncertainty further highlights its limitation, and adding r_E produces slightly but not substantially higher success rates. In comparison, ROMAN's success rates drop slightly when compared with previous levels of noise, but it still retains considerably higher degrees of resilience, highlighting the value of avoiding naively imitating demonstrations.

We conclude that the proposed HHL approach is advantageous in overcoming increasing exteroceptive uncertainties and the complexities associated with longer-time-horizon sequential tasks. Further, in Results, we demonstrate that the HHL architecture of ROMAN dynamically adapts to situations that were not encountered in the demonstrated sequence, and extends beyond the imitated behaviour during training. This is attributed to ROMAN's balance between exploitation and exploration.

Effects of demonstrations

Finally, we compared the effect of different numbers of demonstrations on the overall performance of ROMAN. We analyse the effects of $N = 7$, $N = 21$ and $N = 42$ demonstrations on the success rates across all scenarios for the MN. Our results in Table 6 show that a relatively small number of demonstrations for the MN ($N = 21$, which corresponds to only $N = 3$ demonstrations for each of the seven subtasks) is sufficient to give a reasonable success rate. Doubling the number of demonstrations to $N = 42$ yields higher success rates than for $N = 21$, yet the difference is marginal. A one-shot demonstration of each scenario (that is, $N = 7$) did not yield acceptable success rates during complex sequences as shown in S4–S7. More details regarding the demonstrations can be found in Supplementary Table 7.

Adaptation to recover from local minima

We observed that occasionally experts could fail in retaining a firm grasp, resulting in dropping a grasped object. As shown by the success rates, this occurred fairly infrequently and was primarily limited to experts with picking tasks. Further evaluation found that, when such rare expert-level failures occurred, the MN began to recognize the subtask state and gradually learned a new weight assignment until the tasks were successful. The use of the HHL approach, balancing exploitation and exploration, enabled a positive adaptation of the learning agent to commence a regrasp procedure, as shown in Fig. 2a,b.

Moreover, the MN in ROMAN learns the ability to recover from local minima by rapidly switching experts when it is necessary to do so. During the sequential activation of the seven experts, the robot gripper could occasionally become stuck under the cabinet while retrieving the rack. During such cases, the MN would activate other experts to alter the trajectory and move the gripper away from the cabinet until it was collision free, and then recommence the task successfully, as shown in Fig. 2c. This result highlights the value of combining the advantages of IL and RL paradigms and leveraging intrinsic and extrinsic rewards, resulting in a robust performance in cases not encountered in the demonstrations. Examples can be found in Supplementary Video 1.

Table 3 | Experimental results of ROMAN: success rates are compared between a single NN and ROMAN in two dimensions with five experts

Preliminary version of single NN versus ROMAN on case scenarios					
	S1	S2	S3	S4	S5
$\sigma=\pm0.5$ [cm]	Push	+ Lift	+ Pick and insert	+ Pick and drop	+ Pull
Single NN	0.997	0.841	0.699	0.591	0.565
ROMAN	0.993	0.995	0.982	0.971	0.974

Results stem from 1,000 trials for each individual cell. Noise level is indicated in the leftmost column. Identical numbers of demonstrations, network settings and hyperparameters were used to retain consistency and conduct fair comparisons. A total of $N=20$ demonstrations ($t\approx5$ min) were provided for each expert and a total of $N=35$ demonstrations ($t\approx20$ min) for the gating network. A total of $N=100$ demonstrations were provided to the single NN ($t\approx64$ min). It should be noted that $N=35$ demonstrations for the gating network corresponds to seven demonstrations for each of the five derived case scenarios.

t-SNE analysis of the similarity of sequences

To qualitatively study the MN's ability to activate the necessary expert activation on the basis of its observations, we conducted dimensionality reduction via t-SNE. This allowed us to evaluate the similarities in the observations of the MN and its ability to distinguish between different scenarios. The t-SNE plots are shown in Fig. 3a,b.

First, we conducted a t-SNE on the MN observations at the beginning of each scenario to analyse similarities between the MN observations in different scenarios. As shown in Fig. 3a, scenarios S1–S7 differ to a great degree, and S3, S4 and S5 present a slight overlap with each other because the state vectors between these three are relatively similar. This suggests why the MN may not always activate the correct sequence, particularly at the beginning of a sequence when the end effector's start position is randomized (as opposed to being the ending position of a previous subtask), leading to slightly lower success rates.

Second, we also conducted a t-SNE on the MN observations of each separate activation for every scenario studied. Figure 3b reveals the similarities in the MN observations throughout different expert activations in each of the seven case scenarios. By sampling within the sequence of actions, we obtain a low-dimensional projection of the trajectory of the MN observation vectors during the expert activations. In essence, this is due to the change in the spatial states of the objects in the scene and the end effector being in motion during the sequence of actions.

Overall, Fig. 3b shows no notable overlaps between the activations of the different experts within each scenario, and suggests that the MN is capable of distinctly activating experts during the subtask completion. Thus, regarding the decreased performance for S3, S4 and S5 observed in Tables 2–6, on the basis of the slight overlap between MN observations analysed in Fig. 3a, we conclude that the failures that account for the slight drop in performance occurred at the beginning of the sequences due to the randomized initialization.

Discussion

Results of ROMAN and its implications

The hierarchical task decomposition of ROMAN allows for task-level experts to be trained to achieve robust performance in considerably complex sequential tasks. Hence, it enables the MN to focus on orchestrating these high-level experts, rather than low-level skills, thereby offloading unnecessary complexity from the MN. Our results show that ROMAN can orchestrate notably more complex sequential tasks of longer time horizons and higher dimensionalities than similar work in physics-based manipulation[8,44,45].

Moreover, ROMAN's HHL architecture (Fig. 4) achieved successful adaptation to non-encountered scenarios and recovery from local minima that were not explicitly demonstrated. Hence, the results suggest that, although IL is effective in providing a baseline, achieving a balance between imitating the demonstrations and maximizing the extrinsic RL reward through random exploration is crucial for successful adaptation beyond the demonstrated behaviours. This balance between exploration and exploitation provided by ROMAN also shares common ground with biological studies[27,39].

Finally, results show that ROMAN's central MN was able to solve the most complex and longest-horizon sequential manipulation tasks skilfully. Further investigation also found a performance drop in some of the tasks with lower complexity, such as S3–S5, compared with more complex ones such as S6 and S7. The t-SNE analysis concluded that this is primarily due to the difficulty for the MN to differentiate between those states at randomized initialization of tasks. Future work can explore more sensory feedback to differentiate ambiguous cases, or design a 'memory' mechanism by expanding the observation with history states.

Future work

Future work includes extending ROMAN to higher-dimensionality problems, such as multiexpert HL and bi-manual manipulation. Moreover, to enable real-world deployment in future work, a vision system for exteroceptive information would be needed to predict object poses—for example, using AprilTags, or segmenting/detecting objects using RGB/RGB-D cameras. Additionally, a dynamic grasping controller that incorporates force control could further enhance the grasping performance.

Methods

ROMAN is characterized by an HHL approach. In this architecture, multiple experts specialize in diverse and fundamental types of manipulation tasks that are activated, in the correct sequence, by a primary gating network, the MN. The validation of ROMAN will by definition be among different types of manipulation tasks commonly seen in robotics and physics-based interactions.

System overview

We validate our architecture in a complex medical laboratory setting, to highlight our approach in a setting where manipulation typically consists of (1) careful handling of small objects, (2) the necessity to perform multiple tasks and (3) the correct sequence of tasks to complete a long and complex end goal. The construction of the environment was done in such a way as to derive as many subtasks as possible and validated our method. We used the seven-degree-of-freedom Franka Emika robot in simulation with its default gripper in 3D space, based entirely on physics-based interactions with the environment. The system overview including the simulation environment and the overall depiction of the ROMAN framework are shown in Fig. 4. An architecture overview of the incorporated NNs in the ROMAN framework, including their individual states, actions, number of demonstrations and training time, is given in Table 1. More details of the system and simulation overview, and incorporated software tools[55], including the general apparatus, can be found in the Supplementary Notes and more specifically Supplementary Note 1.

Vision system

As part of our preliminary investigation, we implemented a vision system using an RGB camera in the simulation to predict the poses of the different objects of interest (OIs). The vision system implements an object detection and pose estimation module based on the VGG-16 backbone architecture[56]. The system was initialized with pretrained weights on the ImageNet dataset and fine-tuned using a custom dataset, which was created by capturing the OIs from the simulated environment, including both the segmentation and labelling of the OIs. The output of the network predicted the poses of all OIs, specifically their 3D positions.

Table 4 | Experimental results of ROMAN: success rates across all seven scenarios, between a single NN and ROMAN in three dimensions

Single NN versus ROMAN on case scenarios							
σ=±0.5 [cm]	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**
Single NN	0.997	0.981	0.583	0.032	0.028	0.000	0.000
ROMAN	0.973	0.975	0.817	0.959	0.852	0.960	0.952

Results stem from 1,000 trials for each individual cell. Noise level is indicated in the leftmost column. Identical numbers of demonstrations, network settings and hyperparameters were used to retain consistency and conduct fair comparisons. A total of $N=20$ demonstrations were provided to each agent in ROMAN and a total of $N=42$ demonstrations to the MN. A total of $N=140$ demonstrations were provided to the single NN in three dimensions ($t\approx132$ min).

The rationale for testing with a camera set-up was to validate ROMAN's robustness in a realistic setting, where pose prediction errors and visual occlusions naturally occur. When the target objects were occluded, the last known position was provided to the gating network. Since the pretrained object detection module from the vision system attained variable levels of positional error[56], we simulated increasing levels of Gaussian-distributed noise to all exteroceptive observations of all NNs, so as to further test ROMAN's capabilities besides its robustness to a vision system, which is in line with related work[8,45]. Overall, by introducing exteroceptive uncertainties, we can further assess the resilience of our framework and highlight the importance of a hybrid learning approach within a hierarchical architecture for solving complex sequential tasks.

Learning approach and preliminaries

We make use of two IL algorithms, GAIL[36] and BC[57]. These two algorithms, coupled with the RL algorithm PPO[20], allowed us to successfully and robustly imitate complex daily activity tasks for the purpose of autonomous robotic operation and physics-based interactions entailing multiple tasks. The hybrid learning procedure used for both the expert NNs and the MN in ROMAN is illustrated in Fig. 4a, while Fig. 4b depicts the hierarchical framework formation. In particular, the training procedure is composed of two stages: in stage one, the policy is warm-started using BC; in stage two, the policy is updated via the PPO algorithm with r_E and r_I stemming from the environment (RL) and from the discriminator network (GAIL), respectively.

BC (warm-starting the policy). Foremost, to warm-start the policy, we used BC for a given number of initial epochs. The cutoff point for BC was determined via preliminary investigations and training sessions on the performance of the policy and the complexity of the sequential tasks. Notably, the cutoff point of BC was increased when transitioning from the 2D to 3D version of ROMAN to account for the increased complexity. We avoided using exclusively BC throughout the training process, so as to allow the agent to explore further samples and improve upon demonstrated behaviours, while keeping the demonstration dataset small[12,36]. This is due to BC being limited in its ability to generalize to out-of-distribution states, and thus is restricted to the trajectories seen in the provided demonstrations[36,58]. Most notably, this can lead to drifting errors when the agent encounters new trajectories outside those in the demonstrations[36,59]. In line with previous work concerned with robotic manipulation, sole dependence on BC should be avoided, and instead a viable alternative is to add a reward term when computing a separate RL gradient that corresponds to the BC loss[45]. In our work, using a dataset of state and action transitions s_t^{d}, a_t^{d} provided by the demonstrator, we implement BC by training an NN policy $\pi(s_t) = a_t$ using supervised learning to minimize the mean squared error loss between a_t^{d} and a_t for the demonstration dataset.

GAIL (commenced after BC and active throughout). To effectively match human demonstration data over a period, also known as a horizon, we made use of inverse RL and, in this case, GAIL[36]. GAIL was used after BC's cutoff point, at which GAIL commenced and was active throughout training to attempt to minimize the divergence between the agent's policy and that of the demonstrator. However, GAIL was not directly used to update policy parameters; we instead make use of a proxy imitation reward signal obtained by GAIL, described further in this section.

This is achieved by sampling a set of expert (τ_{E}) and agent (τ_{A}) trajectories of states and actions (s_t, a_t). The expert trajectories are sampled from a given demonstration dataset while the agent trajectories are sampled from a generative model also known as the generator (G). The generator, however, instead of being rewarded solely by the environment, is rewarded by a scalar score provided by the discriminator (D), implemented as a separate NN. In this process, the discriminator attempts to differentiate between the expert and agent trajectories, rewarding the generator if the divergence between these trajectories decreases. The discriminator is also trained to become 'stricter', resulting in the generator, for example, agent, improving its performance at imitating and converging towards the behaviour that was demonstrated by the human expert. This can be formulated as follows:

$$E_{\tau_{\mathrm{E}}}[\nabla \log(D(s_t, a_t))] + E_{\tau_{\mathrm{A}}}[\nabla \log(1 - D(s_t, a_t))] \tag{1}$$

where $E_{\tau_{\mathrm{E}}}$ and $E_{\tau_{\mathrm{A}}}$ represent the expert and agent trajectories from the training, which are represented as inputs to the discriminator network (D). The discriminator outputs a continuous value between 0 and 1, with a value closer to 1 meaning that the agent or generator is resembling a trajectory closer to that of the expert's, essentially minimizing the divergence and maximizing the imitation. Hence, D can be used as a reward signal to train G to mimic the expert's demonstrated data. Moreover, to allow the agent to further explore additional actions that can lead to improved performance when compared with what was demonstrated, we modify the above formulation for the discriminator to use only the states (s_t) and not the actions (a_t) of the demonstrated trajectories. In turn, this leads to increased exploration, which should encourage behaviours beyond those encountered in a demonstrated sequence when coupled with RL (more details are described in Results and Discussion).

Consequently, we reformulate Equation (1) as with ref. 60:

$$E_{\tau_{\mathrm{E}}}[\nabla \log(D(s_t))] + \mathrm{E}_{\tau_{\mathrm{A}}}[\nabla \log(1 - D(s_t))]. \tag{2}$$

Sampling only the states for GAIL allowed us to be less restrictive in terms of imitation. Discriminating against both states and actions between the demonstrator and the expert, as with the original formulation of GAIL[36], would have potentially led to disallowing further exploration by the agent of other actions, which may in actuality lead to better adaptation based on the state space and avoid a 'naive' copying of identical imitation.

The result of using the above two IL algorithms translated into a considerably reduced necessary dataset, compared with related work to train the agents successfully in complex long-horizon sequential tasks[12,44].

Table 5 | Experimental results of ROMAN: success rates across all seven scenarios, between different comparisons of HRL, HIL and their combinations, used to train ROMAN

Algorithm comparison in ROMAN							
σ=±0.5 [cm]	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**
HRL:RL	0.000	0.000	0.000	0.000	0.000	0.000	0.000
HIL:GAIL	0.980	0.468	0.559	0.012	0.003	0.001	0.000
HIL:BC	0.986	0.978	0.786	0.660	0.525	0.722	0.760
HHL:RL,GAIL	0.981	0.468	0.570	0.009	0.005	0.006	0.004
HHL:RL,BC	0.995	0.897	0.841	0.683	0.492	0.754	0.774
ROMAN's †	0.973	0.975	0.817	0.959	0.852	0.960	0.952
σ=±1.0 [cm]	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**
HIL:BC	0.995	0.990	0.712	0.573	0.474	0.563	0.632
HHL:RL,BC	0.996	0.895	0.881	0.766	0.562	0.696	0.729
ROMAN's †	0.977	0.990	0.798	0.946	0.776	0.933	0.939
σ=±2.0 [cm]	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**
HIL:BC	0.838	0.678	0.609	0.205	0.190	0.111	0.075
HHL:RL,BC	0.947	0.841	0.725	0.442	0.363	0.246	0.100
ROMAN's †	0.967	0.986	0.737	0.837	0.753	0.820	0.815

Results stem from 1,000 trials for each individual cell. Noise levels are indicated in the leftmost column. Identical numbers of demonstrations, network settings and hyperparameters were used to retain consistency and conduct fair comparisons. BC: supervised learning on the demonstration dataset. GAIL: use of IL r_I provided to PPO. RL: use of task r_E provided to PPO. ROMAN's †: default HHL approach combining BC, IL (via r_I) and RL (via r_E). Tested on σ=±0.5 cm noise, with up to σ=±1.0 cm and σ=±2.0 cm for algorithms scoring high. Where BC or GAIL is used, the same number of demonstrations (N=42) was employed.

RL (exploration beyond imitation). In addition to the IL approaches mentioned above, we also made use of a small task-related extrinsic reward signal. We use extrinsic rewards to provide a small contribution towards the final policy to avoid exclusive dependence on pure imitation. As described below, we use intrinsic (from IL) as well as extrinsic task-related rewards to update the policy, with the IL reward being scaled by the highest weight and by extent being the main learning signal provider. Most notably, this HHL architecture showed the ability to adapt to new cases that were not encountered during demonstrations, and resilience in the presence of sensor uncertainty. Specifically, this allowed ROMAN to recover from local minima during the most complex sequence activation of experts, even when the sequence is not activated precisely or errors occur in individual experts. We chose PPO as our RL algorithm because it is robust and flexible across various hyperparameter settings.

Denoting our policy π_θ as an NN parameterized by weights θ, the PPO update at step k is given by

$$\theta_{k+1} = \arg\max_{\theta} \mathbb{E}_{s,a\sim\pi_{\theta_k}}\left[L(s,a,\theta_k,\theta)\right] \qquad (3)$$

with a clipped loss function $L(s, a, \theta_k, \theta)$ that has a surrogate term, a value term and an entropy term[20].

Integration of BC, GAIL and RL. To learn to solve long and complex sequential tasks using limited demonstration data, we integrate a set of algorithms for an effective balance between exploitation and exploration. While using BC, we perform supervised learning on the policy using the demonstrations as a dataset: that is, policy updates are driven by the mean squared error loss on the demonstration dataset. While using GAIL and or RL, we use PPO as the general-purpose algorithm to perform policy updates. We then combine these methods by using different reward terms for r_I and r_E, where intrinsic rewards are provided by the discriminator score from GAIL, and extrinsic rewards are provided by the environment as per the RL formalism.

Regarding GAIL, as mentioned above, we modify the original framework to use only states in the discriminator, instead of states and actions, hence making use of Equation (2). We define the intrinsic reward term as $r_I = -\log(1 - D(s_t))$, where $D(s_t) \in (0, 1)$, and acts as a proxy reward term that can be used by PPO to maximize the GAIL objective. When training with GAIL and RL, we use a linear combination of reward terms such that $r = r_I w_I + r_E w_E$, with w_I and w_E fixed scaling parameters for intrinsic and extrinsic rewards, respectively. Our HHL control policy focuses more on imitation, that is, on the intrinsic compared with the extrinsic rewards: the r_I are several magnitudes larger than the r_E ($w_I > w_E$). Using the latter reward combination, the returns are computed as the discounted sum of rewards, and are used for the PPO update on the policy as in Equation (3).

ROMAN's robustness is attributed to the above-employed hybrid learning architecture and in particular the combination of (1) using BC up to a given epoch for warm-starting policy optimization, (2) thereafter using the intrinsic reward provided by GAIL to further minimize the divergence of the agent and that of the expert demonstrator and finally (3) the addition of an extrinsic reward term from the RL paradigm to allow the agent to explore further and beyond what was demonstrated.

The individual NN architecture of each expert and the MN (the gating network), and the hierarchical architecture, are depicted in Fig. 4a,b, respectively. Figure 4b illustrates the hierarchical formation of ROMAN and, more specifically, that the exteroceptive information provided to each NN from the environment is determined by the objective of each expert and the relevance of that information for the successful completion of the given subtask. In contrast, the MN observes the entirety of the environment.

Demonstration acquisition and settings

All demonstrations were provided via keybindings from a generic keyboard, as shown in Fig. 4a. The keyboard was used to provide two levels of demonstrations. First, demonstrations were provided to the expert NNs, with keybindings corresponding to the velocity control of the robotic end effector and the binary state of opening or closing the gripper. The expert NNs shared identical actions and specialized

Table 6 | Experimental results of ROMAN: success rates based on the number of demonstrations provided to the MN

Demonstration comparison *N*=7, 21 and 42 on case scenarios							
Total demo no.	S1	S2	S3	S4	S5	S6	S7
N=7 (*t*≈7 min)	0.775	0.876	0.680	0.378	0.360	0.008	0.005
N=21 (*t*≈25 min)	0.994	0.921	0.718	0.945	0.903	0.929	0.958
N=42 (*t*≈42 min)	0.973	0.975	0.817	0.959	0.852	0.960	0.952

Results stem from 1,000 trials for each individual cell. Identical numbers of demonstrations, network settings and hyperparameters were used to retain consistency and conduct fair comparisons. The total number of demonstrations is divided by the number of scenarios. Consequently, *N*=7, *N*=21 and *N*=42 correspond to one, three and six demonstrations per case scenario, respectively. Evaluated on σ=±0.5 level of noise.

in different manipulation skills. Second, given the pretrained expert NNs, a demonstrated sequence for the MN was provided via a set of different keybindings corresponding to the weight assignment of the incorporated experts in the hierarchical architecture. Therefore, the expert demonstrations were specific to each individual expert's specialized skill and goal, which allowed these pretrained networks to be coordinated by the MN's demonstrated sequence for the sequential activation of the task.

Two cameras in an orthographic projection were rendered onto a 2D monitor, visually displaying the environment from an upper and side-view perspective that allowed the human expert to observe the task and behaviour. In such a simulated environment, the determination of depth-associated distances is rendered easier for the human demonstrator, as shown by previous work[1,4,61].

These demonstrations were used to warm-start the policy via BC and for the discriminator of GAIL in the form of an intrinsic reward to the PPO algorithm. A total of $N = 20$ demonstrations were provided to pretrain the ROMAN expert NNs, and a total of $N = 42$ demonstrations were provided to the MN, corresponding to $N = 6$ for each of the seven sequential scenarios. Our technical approach of using a keyboard for generating demonstration data and imitating trajectories for the expert NNs and the MN, as well as the corresponding technical implementations, was not derived from our previous work or other published works.

Task

The physics engine NVIDIA PhysX allowed us to devise numerous tasks all containing physical properties and advanced physical characteristics such as hinges, linearly moving objects and spring joints. The full task is visually illustrated in Fig. 1, with the full sequence decomposed into its relevant subtasks in Fig. 3d.

The task was conceived and inspired by a medical laboratory setting, where frequently encountered manipulation tasks often involve a varying and flexible number of sequences of differing types of subtasks. The objective here was to retrieve a small vial, insert it into a rack and push them all together onto a conveyor belt. Within this workflow, we further derived additional subtasks while ensuring their interdependence. All derived tasks are common in robotic manipulation and physics-based interactions[5]. With a total of seven experts as in Fig. 1, we derived seven sequence activation cases, referred to as scenarios. The numbering of scenarios also indicates how many experts are involved, as each sequence builds upon the previous one by adding a new task to it. Finally, the episode terminated once either (1) the button next to the conveyor belt was pushed, (2) the maximum step count for the episode was reached or (3) the end effector deviated too far from the centre of the scene.

Expert network characteristics and architecture

A complex, long-horizon sequential task in full 3D space is decomposed into fundamental and high-level types of manipulation skill, henceforth referred to as experts. This allowed the validation of the robustness of the architecture over increasing complexity, uncertainty and dimensionality. The manipulation experts are derived with diverse and distinct specialized skills to cover a broad range of common tasks in real-life and robotic manipulation[3,38]. We ensured that these experts were not too closely interrelated to one another, thereby offering greater versatility and flexibility while used in combination. The total number of trained expert NNs is seven, as shown in Fig. 1 and listed below.

- **Pull Open (open drawer)** [π_{Pull}]: expert responsible for pulling a linearly moving object, such as a sliding drawer.
- **Pick and Drop (unbox)** [π_{PickDrop}]: expert responsible for picking and dropping an object without regard to height offset. This is commonly seen when removing the lid or the cover of a disposable box to retrieve an OI.
- **Rotate Open (open cabinet)** [$\pi_{\mathrm{RotateOpen}}$]: expert responsible for rotating a door handle configured around a single axis, a very common scenario seen when opening a cabinet or rotating drawer.
- **Pick and Place (place rack)** [$\pi_{\mathrm{PickPlace}}$]: expert responsible for picking and placing an object carefully (with zero or close to minimal height drop).
- **Pick and Insert (insert vial)** [$\pi_{\mathrm{PickInsert}}$]: expert responsible for picking and inserting an object with high precision in a particular docking target location.
- **Push (push rack and vial)** [π_{Push}]: expert responsible for pushing an object over a surface.
- **Push Button (push button)** [π_{Button}]: expert responsible for pushing a human-made switch or button.

Action space. All aforementioned experts are listed in the form of high-level abstract manipulation type (specific task on validated environment). All experts shared identical actions, including full end-effector velocities (α_1, $\pm\upsilon_x$; α_2, $\pm\upsilon_y$; α_3, $\pm\upsilon_z$), as well as controlling the gripper state (α_4, $f(\pm x_g)$). Sharing the same action space across experts is relevant to highlight the value of our proposed hierarchical framework, as expert specialization is not aided by constraining the actions available to each expert to those that are only relevant for its respective specialization.

State space. The state space of each expert was identical for the proprioceptive and sensory states; however, it differed for the exteroceptive states depending on each specialized manipulation skill and the relevant information from the environment for the successful completion of each individual task. Consequently, the exteroceptive states were decided on the basis of the nature of each expert's specialized skill and end goal. This allowed each NN to focus only on its own core exteroceptive information relevant to its subtask, and omit non-relevant ones—as seen from a neuroscientific perspective, whereby during the human decision-making process the relevance of information during a motor task is determined and specified[62]. The state and action spaces, including the demonstration settings and training times for each expert and the MN, are detailed in Table 1.

High-level task decomposition. The derived experts were composed in such a way as to allow a high-level task decomposition, thereby offloading the central MN from combining a large number of low-level action-based experts that can otherwise be solved by a single subtask-based expert. This was made possible by virtue of the employed hybrid learning procedure in the hierarchical architecture of ROMAN, which incorporates and orchestrates multiple NNs specialized in subtasks to efficiently and effectively solve complex manipulation tasks over a long time horizon.

In contrast, most related work decomposed manipulation experts into rather basic action-based primitives or action-level skills[12,44]. While this allows for the derivation of more abstract cases, it does limit the potential of a hierarchical model. In particular, a decomposition of low-level action-based skills prevents, to a great extent, the gating network from learning high-level scene understandings or solving complex sequences, as it focuses more on composing skills such as picking and placing, which can be instead solved by one single expert. For example, in ref. 44, the skill of the picking and placing task was learned using a three-expert hierarchical architecture composed of (1) approaching, (2) manipulating and (3) retracting. ROMAN's framework shows that, by virtue of the employed hybrid learning approach, the derivation of picking and placing as a single high-level expert is made possible. This is how our HHL architecture overcomes such limitations by deriving experts specialized in high-level subtask-based manipulation skills, offloading the MN in turn from lower-level skill supervision.

Moreover, each single task-level expert trained via the employed hybrid learning procedure has its own inherent robustness in facing new states during the exploration of the RL process. This allowed the gating network to be trained more effectively in solving highly complex sequences over long time horizons, without the need to learn how to recombine primitive action-based experts to achieve a subtask.

From the results, we observed that when the MN was switching between the different experts there was a possibility of dropping the object when suddenly switching from any expert involved with picking to another. This is due to the limited control interface of the gripper, which provides only binary commands for opening and closing. To compensate for this, a dead zone (DZ) is introduced to account for the expert switching process. This relationship is shown as

$$\mathrm{DZ}\,(x_g) = \begin{cases} \text{close} & \text{if } x_g \in [-1.0, -0.9], \\ \text{remain the same} & \text{if } x_g \in (-0.9, 0.9), \\ \text{open} & \text{if } x_g \in [0.9, 1.0]. \end{cases} \quad (4)$$

Hence, the DZ ($\in (0, 1)$) implementation improved the overall stability of grasping, by only switching the gripper action to open or close when x_g goes beyond the zone of (−0.9, 0.9). As a possible future work, one could potentially substitute the DZ implementation and enhance grasping control by incorporating a dynamic controller with force control or tactile sensing to render grasping more stable and reliable.

MN characteristics

The MN acts as a master control policy, overseeing the expert NNs and assignING weights ($\in (0, 1)$) to them. The final output is defined as the sum of these weighted actions:

$$\sum_{i=1}^{m} \sum_{j=1}^{n} \alpha_i w_j \quad (5)$$

whereby the number ($m = 4$) of all actions (α_i) of each expert is controlled by a set of weights (w_j) corresponding to the total number ($n = 7$) of experts in the hierarchy. One of the main issues in assigning the weights is to ensure the sum of all weights does not exceed unity, which can lead to unwanted behaviour, most notably torques and forces going beyond the robot's capabilities.

Hence, we normalize the sum of weights assigned by the MN to activate experts, using a normalized exponential function, that is, softmax, which provides a probability distribution to better isolate the expert activation during long sequences. This is represented as

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad \text{for } i = 1, \ldots, K \text{ and } \mathbf{z} = (z_1, \ldots, z_K) \in \mathbb{R}^K \quad (6)$$

where σ is the softmax function and $\mathbf{z}$ is the input vector, as a function of e^{z_i}, denoting the standard exponential for each input, divided by the sum of all inputs K. In our case, the input vector is represented as a weight vector, with each element representing the weight of every single expert, with a sum equal to $K = 7$, representing all seven distinct experts.

The observation space of the MN contains the union of all of the observation spaces of each individual expert. Consequently, the observation spaces of the experts consist of the environment states that are relevant to their task, while the MN essentially observes the entirety of the relevant subtasks to better distinguish which expert should be activated at which time step. Figure 4 depicts the overall ROMAN framework, highlighting the MN as a gating mechanism that centrally governs the control policy in the HHL control framework.

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

The full evaluation data are publicly available and can be accessed at https://github.com/etriantafyllidis/ROMAN_Data. The data can be downloaded as a compressed file (.zip) and consist of comma-separated values formatted files for each evaluated scenario, including success rates for subtasks and end-goal sequences. A ReadMe file is included with further details on the number of demonstrations, noise perturbations and other relevant information. Microsoft Excel Version 2305 Build 16.0.16501.20074 64 bit was used to interpret, read and summarize the statistical results.

Code availability

The code of ROMAN is publicly available at https://github.com/etriantafyllidis/ROMAN (ref. 63).

References

1. Triantafyllidis, E. & Li, Z. The challenges in modeling human performance in 3D space with Fitts' law. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems, CHI EA '21* 56 (Association for Computing Machinery, 2021).
2. Ashe, J., Lungu, O. V., Basford, A. T. & Lu, X. Cortical control of motor sequences. *Curr. Opin. Neurobiol.* **16**, 213–221 (2006).
3. Ortenzi, V. et al. Robotic manipulation and the role of the task in the metric of success. *Nat. Mach. Intell.* **1**, 340–346 (2019).
4. Triantafyllidis, E., Mcgreavy, C., Gu, J. & Li, Z. Study of multimodal interfaces and the improvements on teleoperation. *IEEE Access* **8**, 78213–78227 (2020).
5. Billard, A. & Kragic, D. Trends and challenges in robot manipulation. *Science* **364**,1149 (2019).
6. Tee, K. P., Cheong, S., Li, J. & Ganesh, G. A framework for tool cognition in robots without prior tool learning or observation. *Nat. Mach. Intell.* **4**, 533–543 (2022).
7. Davchev, T. et al. Wish you were here: hindsight goal selection for long-horizon dexterous manipulation. In *International Conference on Learning Representations* (ICLR, 2022).
8. Fox, R., Berenstein, R., Stoica, I. & Goldberg, K. Multi-task hierarchical imitation learning for home automation. In *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)* 1–8 (IEEE, 2019).
9. Flanagan, J. R., Bowman, M. C. & Johansson, R. S. Control strategies in object manipulation tasks. *Curr. Opin. Neurobiol.* **16**, 650–659 (2006).
10. Triantafyllidis, E., Yang, C., McGreavy, C., Hu, W. & Li, Z. in *AI for Emerging Verticals: Human–Robot Computing, Sensing and Networking* (eds Shakir, M. Z. & Ramzan, N.) 63–100 (IET, 2020).

11. Zhang, H., Ye, Y., Shiratori, T. & Komura, T. Manipnet: neural manipulation synthesis with a hand–object spatial representation. *ACM Trans. Graph.* **40**, 121 (2021).
12. Zhang, T. et al. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)* 5628–5635 (IEEE, 2018).
13. Chebotar, Y. et al. Closing the sim-to-real loop: adapting simulation randomization with real world experience. In *2019 International Conference on Robotics and Automation (ICRA)* 8973–8979 (IEEE, 2019).
14. Lee, M. A. et al. Making sense of vision and touch: self-supervised learning of multimodal representations for contact-rich tasks. In *2019 International Conference on Robotics and Automation (ICRA)* 8943–8950 (IEEE, 2019).
15. Schill, M. M., Gruber, F. & Buss, M. Quasi-direct nonprehensile catching with uncertain object states. In *2015 IEEE International Conference on Robotics and Automation (ICRA)* 2468–2474 (IEEE, 2015).
16. Schoettler, G. et al. Deep reinforcement learning for industrial insertion tasks with visual inputs and natural rewards. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* 5548–5555 (IEEE, 2020).
17. Andrychowicz, O. M. et al. Learning dexterous in-hand manipulation. *Int. J. Robot. Res.* **39**, 3–20 (2020).
18. Zhang, H., Starke, S., Komura, T. & Saito, J. Mode-adaptive neural networks for quadruped motion control. *ACM Trans. Graph.* **37**, 145 (2018).
19. Yang, C., Yuan, K., Zhu, Q., Yu, W. & Li, Z. Multi-expert learning of adaptive legged locomotion. *Sci. Robot.* **5**, eabb2174 (2020).
20. Schulman, J., Wolski, F., Dhariwal, P., Radford, A. & Klimov, O. Proximal Policy Optimization algorithms. Preprint at https://arxiv.org/abs/1707.06347 (2017).
21. Haarnoja, T., Zhou, A., Abbeel, P. & Levine, S. Soft Actor–Critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor. *Proc. Mach. Learning Res.* **80**, 1861–1870 (2018).
22. Gu, S. et al. Interpolated policy gradient: merging on-policy and off-policy gradient estimation for deep reinforcement learning. In *Proc. 31st International Conference on Neural Information Processing Systems, NIPS'17* 3849–3858 (Curran, 2017).
23. Koganti, N., Hafiz, A. R., Iwasawa, Y., Nakayama, K. & Matsuo, Y. Virtual reality as a user-friendly interface for learning from demonstrations. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems, CHI EA '18* D310 (Association for Computing Machinery, 2018).
24. Ding, Y., Florensa, C., Abbeel, P. & Phielipp, M. Goal-conditioned imitation learning. In *Advances in Neural Information Processing Systems* Vol. 32 (eds Wallach, H. et al.) 15324–15335 (Curran, 2019).
25. Zaadnoordijk, L., Besold, T. R. & Cusack, R. Lessons from infant learning for unsupervised machine learning. *Nat. Mach. Intell.* **4**, 510–520 (2022).
26. Schaal, S. Learning from demonstration. In *Advances in Neural Information Processing Systems* Vol. 9 (eds Mozer, M. C. et al.) 1040–1046 (MIT Press, 1997).
27. Zador, A. M. A critique of pure learning and what artificial neural networks can learn from animal brains. *Nat. Commun.* **10**, 3770 (2019).
28. Thor, M. & Manoonpong, P. Versatile modular neural locomotion control with fast learning. *Nat. Mach. Intell.* **4**, 169–179 (2022).
29. Goldberg, K. Robots and the return to collaborative intelligence. *Nat. Mach. Intell.* **1**, 2–4 (2019).
30. Levine, S. & Abbeel, P. Learning neural network policies with guided policy search under unknown dynamics. In *Advances in Neural Information Processing Systems* Vol. 27 (eds Ghahramani, Z. et al.) 1071–1079 (Curran, 2014).
31. Mnih, V. et al. Human-level control through deep reinforcement learning. *Nature* **518**, 529–533 (2015).
32. Schulman, J., Levine, S., Abbeel, P., Jordan, M. & Moritz, P. Trust region policy optimization. *Proc. Mach. Learning Res.* **37**, 1889–1897.
33. Mnih, V. et al. Asynchronous methods for deep reinforcement learning. *Proc. Mach. Learning Res.* **48**, 1928–1937 (2016).
34. Pastor, P., Hoffmann, H., Asfour, T. & Schaal, S. Learning and generalization of motor skills by learning from demonstration. In *2009 IEEE International Conference on Robotics and Automation* 763–768 (IEEE, 2009).
35. Ratliff, N., Bagnell, J. A. & Srinivasa, S. S. Imitation learning for locomotion and manipulation. In *2007 7th IEEE–RAS International Conference on Humanoid Robots* 392–397 (IEEE, 2007).
36. Ho, J. & Ermon, S. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems* Vol. 29 (eds Lee, D. et al.) 4572–4580 (Curran, 2016).
37. Ross, S., Gordon, G. & Bagnell, D. A reduction of imitation learning and structured prediction to no-regret online learning. *Proc. Mach. Learning Res.* **15**, 627–635 (2011).
38. Triantafyllidis, E., Hu, W., McGreavy, C. & Li, Z. Metrics for 3D object pointing and manipulation in virtual reality: the introduction and validation of a novel approach in measuring human performance. *IEEE Robot. Autom. Mag.* **29**, 76–91 (2021).
39. Saxe, A., Nelli, S. & Summerfield, C. If deep learning is the answer, what is the question? *Nat. Rev. Neurosci.* **22**, 55–67 (2021).
40. Abbeel, P. & Ng, A. Y. Apprenticeship learning via inverse reinforcement learning. In *Proc. Twenty-First International Conference on Machine Learning, ICML '04* 1 (Association for Computing Machinery, 2004).
41. Finn, C., Levine, S. & Abbeel, P. Guided cost learning: deep inverse optimal control via policy optimization. In *Proc. 33rd International Conference on International Conference on Machine Learning, ICML'16* Vol. 48, 49–58 (JMLR.org, 2016).
42. Le, H. M. et al. Hierarchical imitation and reinforcement learning. *Proc. Mach. Learning Res.* **80**, 2923–2932 (2018).
43. Behbahani, F. et al. Learning from demonstration in the wild. In *2019 International Conference on Robotics and Automation (ICRA)* 775–781 (IEEE, 2019).
44. Marzari, L. et al. Towards hierarchical task decomposition using deep reinforcement learning for pick and place subtasks. In *2021 20th International Conference on Advanced Robotics (ICAR)* 640–645 (IEEE, 2021).
45. Rajeswaran, A. et al. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. In *Proc. Robotics: Science and Systems* (Robotics: Science and Systems Foundation, 2018).
46. Liu, Y., Gupta, A., Abbeel, P. & Levine, S. Imitation from observation: learning to imitate behaviors from raw video via context translation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)* 1118–1125 (IEEE, 2018).
47. Frans, K., Ho, J., Chen, X., Abbeel, P. & Schulman, J. Meta learning shared hierarchies. In *6th International Conference on Learning Representations, ICLR 2018 Conference Track Proc.* https://openreview.net/forum?id=SyX0IeWAW (OpenReview.net, 2018).
48. Merel, J. et al. Hierarchical visuomotor control of humanoids. In *7th International Conference on Learning Representations, ICLR 2019* https://openreview.net/forum?id=BJfYvo09Y7 (OpenReview.net, 2019).
49. Merel, J., Botvinick, M. & Wayne, G. Hierarchical motor control in mammals and machines. *Nat. Commun.* **10**, 5489 (2019).
50. Fox, R. et al. Parametrized hierarchical procedures for neural programming. In *6th International Conference on Learning Representations, ICLR 2018 Conference Track Proc.* https://openreview.net/forum?id=rJl63fZRb (OpenReview.net, 2018).

51. Peng, X. B., Chang, M., Zhang, G., Abbeel, P. & Levine, S. MCP: learning composable hierarchical control with multiplicative compositional policies. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019* (eds Wallach, H. M. et al.) 3681–3692 (Curran, 2019).
52. Mülling, K., Kober, J., Kroemer, O. & Peters, J. Learning to select and generalize striking movements in robot table tennis. *Int. J. Robot. Res.* **32**, 263–279 (2013).
53. Antotsiou, D., Ciliberto, C. & Kim, T. Modular adaptive policy selection for multi-task imitation learning through task division. In *2022 International Conference on Robotics and Automation (ICRA)* 2459–2465 (IEEE, 2022).
54. Ross, S., Gordon, G. & Bagnell, D. A reduction of imitation learning and structured prediction to no-regret online learning. *Proc. Mach. Learning Res.* **15**, 627–635 (2011).
55. Juliani, A. et al. Unity: a general platform for intelligent agents. Preprint at https://arxiv.org/abs/1809.02627 (2018).
56. Tobin, J. et al. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* 23–30 (IEEE, 2017).
57. Torabi, F., Warnell, G. & Stone, P. Behavioral cloning from observation. In *Proc. 27th International Joint Conference on Artificial Intelligence, IJCAI'18* 4950–4957 (AAAI Press, 2018).
58. Reddy, S., Dragan, A. D. & Levine, S. SQIL: imitation learning via reinforcement learning with sparse rewards. In *8th International Conference on Learning Representations, ICLR 2020* https://openreview.net/forum?id=S1xKd24twB (OpenReview.net, 2020).
59. Codevilla, F., Santana, E., Lopez, A. & Gaidon, A. Exploring the limitations of behavior cloning for autonomous driving. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* 9328–9337 (IEEE, 2019).
60. Jeon, W., Seo, S. & Kim, K.-E. A Bayesian approach to generative adversarial imitation learning. In *Advances in Neural Information Processing Systems* Vol. 31 (eds Bengio, S. et al.) 7429–7439 (Curran, 2018).
61. Barrera Machuca, M. D. & Stuerzlinger, W. The effect of stereo display deficiencies on virtual hand pointing. In *Proc. 2019 CHI Conference on Human Factors in Computing Systems* 207 (Association for Computing Machinery, 2019).
62. Wolpert, D. M., Diedrichsen, J. & Flanagan, J. R. Principles of sensorimotor learning. *Nat. Rev. Neurosci.* **12**, 739–751 (2011).
63. Triantafyllidis, E., Acero, F., Liu, Z. & Li, Z. etriantafyllidis/roman: Roman v1.0. *Zenodo* https://doi.org/10.5281/zenodo.8059565 (2023).

Acknowledgements

We thank T. Komura, R. Fisher, Q. Rouxel and F. Christianos for their feedback. We thank R. Wen for providing some of the 3D models used in this work. We also thank V. Andries for proofreading contents of this work. E.T. is supported by the EPSRC CDT in Robotics and Autonomous Systems (EP/L016834/1), and F.A. is supported by the UKRI CDT in Foundational Artificial Intelligence (EP/S021566/1).

Author contributions

E.T. contributed to the conceptualization and implementation of the ROMAN architecture, the simulation set-up, data acquisition and analysis, experimentation and designing the visuals and figures, and authored the manuscript. F.A. contributed to the derivation of the ROMAN architecture, data analysis, t-SNE analysis, types of experiment conducted, visuals and figures, and writing of the manuscript. Z. Liu contributed to the ROMAN architecture, data analysis, types of experiment conducted, visuals and figures, and writing of the manuscript. Z. Li directed the research, provided management across all aspects of this research, supported solution of research and technical problems, edited figures and wrote the manuscript. All authors contributed to the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s42256-023-00709-2.

Correspondence and requests for materials should be addressed to Zhibin Li.

Peer review information *Nature Machine Intelligence* thanks Claudio Coppola, Efi Psomopoulou and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

Reprints and permissions information is available at www.nature.com/reprints.

nature portfolio

Corresponding author(s): Dr. Zhibin Alex Li. (NATMACHINTELL-A22116962B)

Last updated by author(s): Jun 20, 2023

Reporting Summary

Nature Portfolio wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Portfolio policies, see our Editorial Policies and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a	Confirmed	
□	☒	The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement
☒	□	A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☒	□	The statistical test(s) used AND whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
☒	□	A description of all covariates tested
☒	□	A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
□	☒	A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
☒	□	For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted *Give P values as exact values whenever suitable.*
☒	□	For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☒	□	For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
☒	□	Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection	The simulation engine Unity3D (2019.3.0f6) was used as the primary simulation, containing the built-in NVIDIA physics engine (PhysX 4.1) coupled with the PyTorch-based ML-Agents toolkit (0.15.1). State-of-the-art learning algorithms were used in this work, including the reinforcement learning algorithm Proximal Policy Optimization (PPO), Behavioral Cloning (BC) as well as Generative Adversarial Imitation Learning (GAIL). No third-party software was used for the data collection itself. We provide all data (.csv files) stemming from the data acquisition from the experiments, which were subsequently used for the statistical analysis and results. The code of ROMAN is publicly available at https://github.com/etriantafyllidis/ROMAN with the following DOI: https://doi.org/10.5281/zenodo.8059565
Data analysis	Microsoft Excel Version 2305 Build 16.0.16501.20074 64-bit was used to interpret, read and summarise the statistical results.

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Portfolio guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A description of any restrictions on data availability
- For clinical datasets or third party data, please ensure that the statement adheres to our policy

Data sets generated during the current study are made available publicly and can be accessed at: https://github.com/etriantafyllidis/ROMAN_Data

Human research participants

Policy information about studies involving human research participants and Sex and Gender in Research.

Reporting on sex and gender	Not Applicable
Population characteristics	Not Applicable
Recruitment	Not Applicable
Ethics oversight	Not Applicable

Note that full information on the approval of the study protocol must also be provided in the manuscript.

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences ☐ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	The sample size varied based on the baseline evaluation in the Results section. These were N=100, N=1000 and N=10000 depending on the baselines studied. To study the robustness of each individual pre-trained experts in ROMAN, a total of N=10000 trials were conducted per each cell listed in Table 1.b across the increasing Gaussian level of 0.0 to 2.5cm with 0.5cm increments over all seven case scenarios. For the manipulation networks' robustness against increasing Gaussian levels of 0.0 to 2.5cm with 0.5cm increments over all seven case scenarios, a total of N=1000 trials were conducted for each cell listed in Table 1.b. For the manipulation network's vision system, a total of N=100 were tested per cell in Table 1.b over all seven case scenarios. For all results derived in sub-tables in Table 2, each cell stems from N=1000 trials across different levels of Gaussian noise, case scenarios, different dimensions (2D and 3D) as well as different numbers of demonstrations N=7, N=21 and N=42.
Data exclusions	No data was excluded from the study. Data was only updated based on the reviewers recommendation of using a Single NN in 2D and 3D space with N=100 and N=140 respectively. Consequently, subsequent data from the older experiments for N=35 and N=42 were updated as the result of the reviewer's recommendation
Replication	Results can be reproduced by validating the provided data and code by the authors.
Randomization	All models trained and subsequently evaluated were randomised.
Blinding	Blinding was not relevant to this study due to the absence of human participants and the collection of data involved the validation of trained machine-learning models.

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study
☒	☐ Antibodies
☒	☐ Eukaryotic cell lines
☒	☐ Palaeontology and archaeology
☒	☐ Animals and other organisms
☒	☐ Clinical data
☒	☐ Dual use research of concern

Methods

n/a	Involved in the study
☒	☐ ChIP-seq
☒	☐ Flow cytometry
☒	☐ MRI-based neuroimaging