[Put-In-Box Task Generated from Multiple Discrete Tasks by a Humanoid Robot Using Deep Learning]

Presenter: [张子尚 包辰博 盛李杰 张中堂 程耀宇]

[2023/4/28]

Motivation and Main Problem

Further explaination of the title with supporting evidence

High-level description of problem being solved

How to design robots that can combine appropriate actions in a dynamic environment to complete a complex task that comprises multiple subtasks.

- "As the number of situations and task types increase, it becomes extremely difficult to define the features and design appropriate motions for each condition."
- "Robot manipulation methods that can extract features easily, function in unknown situations, and generate and combine various tasks to suit changing work environments are desirable. "

Motivation and Main Problem

Why is the problem important?

- **Its significance towards general-purpose robot autonomy**
- Nowadays, most factories are not fully automated. Typically, in a changing work environment, e.g., small quantity production lines for customizable products, people perform various complicated tasks because robot manipulation using a modeling approach often requires experts to manually extract environmental features for various conditions and plan corresponding motion trajectories. As the number of situations and task types increase, it becomes extremely difficult to define the features and design appropriate motions for each condition.

Motivation and Main Problem

Further explanation of the title with supporting evidence

Technical challenges arising from the problem

- As the number of situations and task types increase, it becomes extremely difficult to define the features and design appropriate motions for each condition.
- Hardcoded robot manipulations are often unstable in uncontrolled environments.

- **The role of the AI and machine learning in tackling this problem**
  - DNNs(Deep neural network) can autonomously reduce high-dimensional data to low-dimensional data for feature extraction [1], and they can be used to classify unknown data from learned data [2]. These characteristics could be applied to extend robot manipulation by autonomously extracting environmental features and increasing generalizability.

AncoraSIR.com

Motivation and Main Problem

Further explanation of the title with supporting evidence

Key insights of the proposed work

- I: **Dimension reduction**
- II: **Autonomously extracting environmental features**
- III: **Generalizability**

Problem Setting

Further explanation of the title with supporting evidence

Problem formulation, key definitions and notations

- **1. Deep neural networks (DNN),** can autonomously reduce high-dimensional data to low-dimensional data for feature extraction, and they can be used to classify unknown data from learned data.
- **2. Recurrent neural networks (RNN),** which are DNNs that use an internal memory to calculate the subsequent output from prior inputs, are used for predictive learning.

AncoraSIR.com

Context / Related Work / Limitations of Prior Work

Further explanation of the title with supporting evidence

Which other papers have tried to tackle this problem or a related problem?

- **The paper's related work is a good start, but there may be others**

- C. Wei, Z. Ji and B. Cai, "Particle Swarm Optimization for Cooperative Multi-Robot Task Allocation: A Multi-Objective Approach," in IEEE Robotics and Automation Letters, vol. 5, no. 2, pp. 2530-2537, April 2020, doi: 10.1109/LRA.2020.2972894.
- G. Yasuda, "Hierarchical and distributed implementation of synchronization and coordination for discrete event multiple robot systems," Proceedings of 2012 IEEE/ASME 8th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications, Suzhou, China, 2012, pp. 302-307, doi: 10.1109/MESA.2012.6275579.
- Hyun-Wook Jo, Jae-Ho Ahn, Jun-Sang Park, Jun-Han Oh and J. -T. Lim, "Task planning for service robots with optimal supervisory control," 2010 IEEE Conference on Robotics, Automation and Mechatronics, Singapore, 2010, pp. 303-308, doi: 10.1109/RAMECH.2010.5513172.

Context / Related Work / Limitations of Prior Work

Further explanation of the title with supporting evidence

Which other papers have tried to tackle this problem or a related problem?

- **What is the key limitations of prior work(s)?**

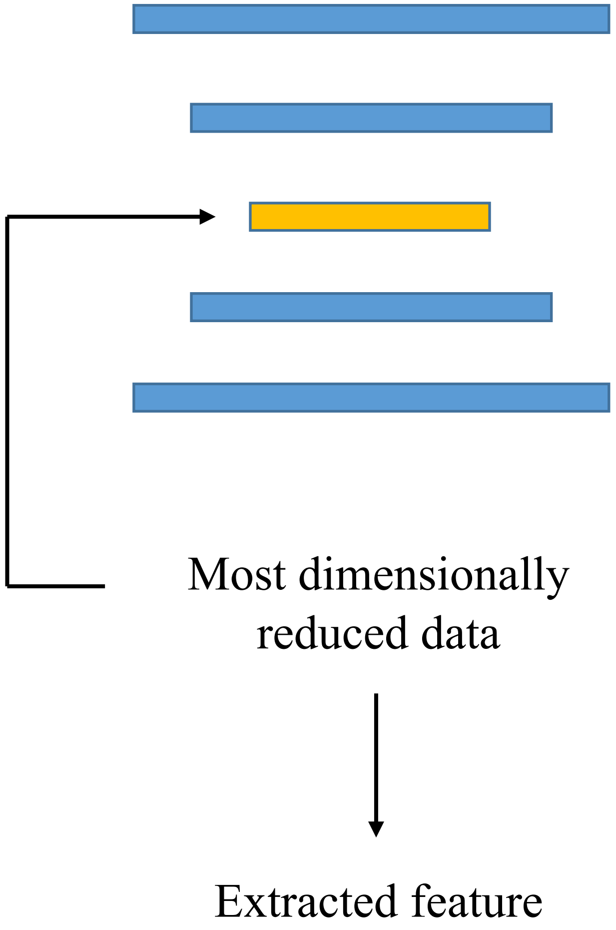
- (1) A previous study [4] focused on training multiple controllers and performing tasks in series; however, that study did not exploit visual inputs. Another study [3] focused on end-to-end visual servoing using RL. In that study, the robot demonstrated several types of task generation; however, the tasks were trained separately, i.e., multiple tasks were not trained simultaneously, and a switching phase was not considered.

- (2) One study employed a low-cost robotic arm to generate multiple tasks using LSTM [7]; however, the transitions between tasks were not considered.

- (3) For robot manipulation using predictive learning, several studies have trained multiple tasks simultaneously [5] [6]. Numerous tasks have been generated using a multimodal DNN that integrates image features with the joint angles of a robot [6]. However, even though these frameworks deal with multiple tasks, they use consumer-grade robots to perform simple tasks, such as moving a box or rolling a ball, and do not consider switching tasks or complex tasks.

Proposed Approach

Convolutional Autoencoder (CAE)



Usage: Extract the image features

- Have features of both an AE and a CNN
- Operate as an identity function using a bottleneck structure, which reduces the dimensionality of the original data and then reconstructs using convolutional and deconvolutional layers
- The pooling layer removed for positional information
- Trained using mean squared error with the optimizer for the Adam algorithm

AncoraSIR.com

Proposed Approach

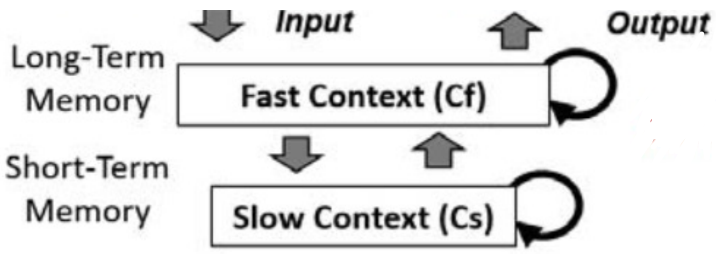
Multiple Timescale RNN

- **Aim:** Simpler and more direct analysis of internal representation
- RNNs can predict the next step from previous steps due to the existence of context layers.

Proposed MTRNN → Fast Context Layer *Cf* (Small time constant) → More info from the current context

Proposed MTRNN → Slow Context Layer *Cs* (Large time constant) → More info from the previous context

Input data at t $\xrightarrow[\text{Train MTRNN}]{\text{MSE and the Adam optimizer}}$ Predicted data at $t+1$



AncoraSIR.com

Proposed Approach

Multiple Timescale RNN

- Similar values for initial and final values of the contexts in order to switch among subtasks

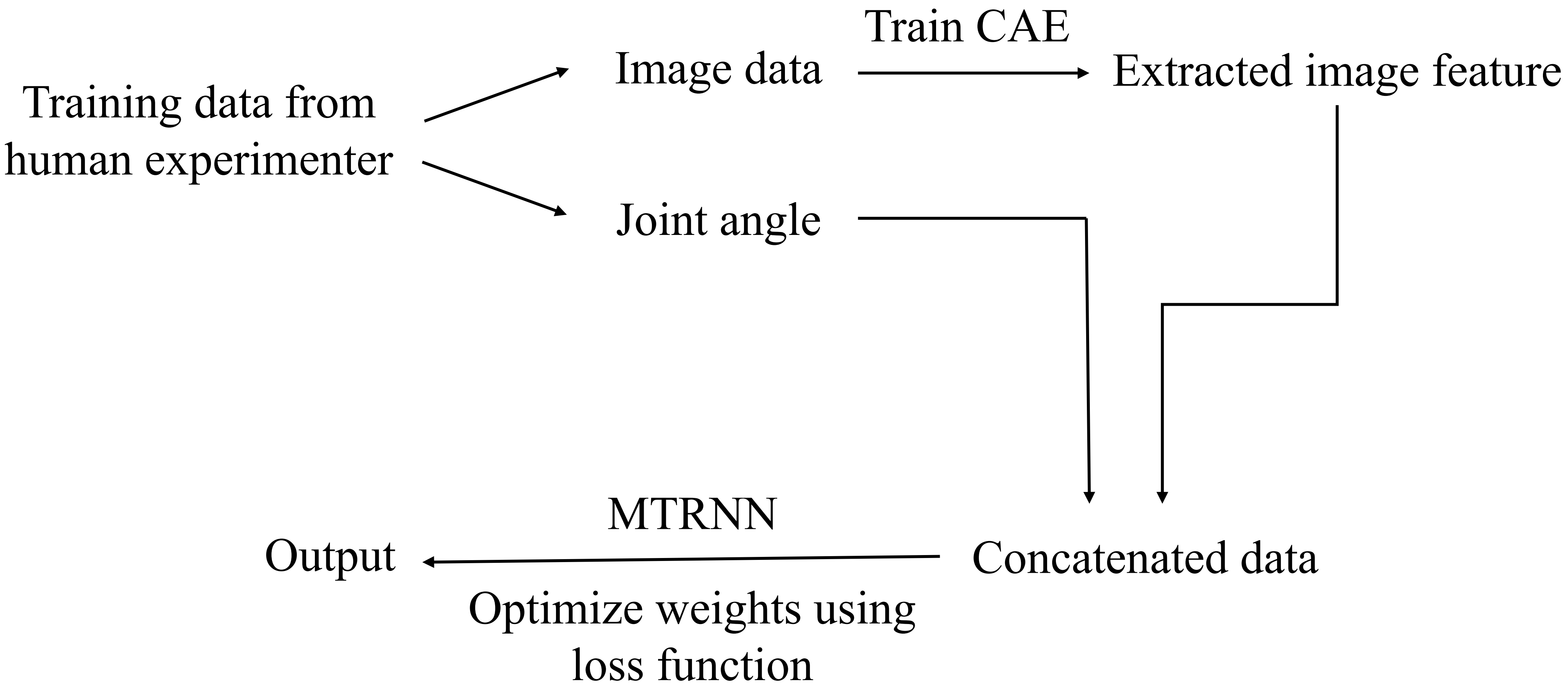
- Constraint: $$loss = \sum_{t=0}^{T} \|(\hat{y}(t) - y(t))\|^2 + \gamma(C(T) - C(0))^2,$$

- Allows the network to form the strong attracting point, the point at which the internal states are likely to converge to and diverge from

AncoraSIR.com

Proposed Approach

Training Phase



AncoraSIR.com

Proposed Approach

Motion Generation

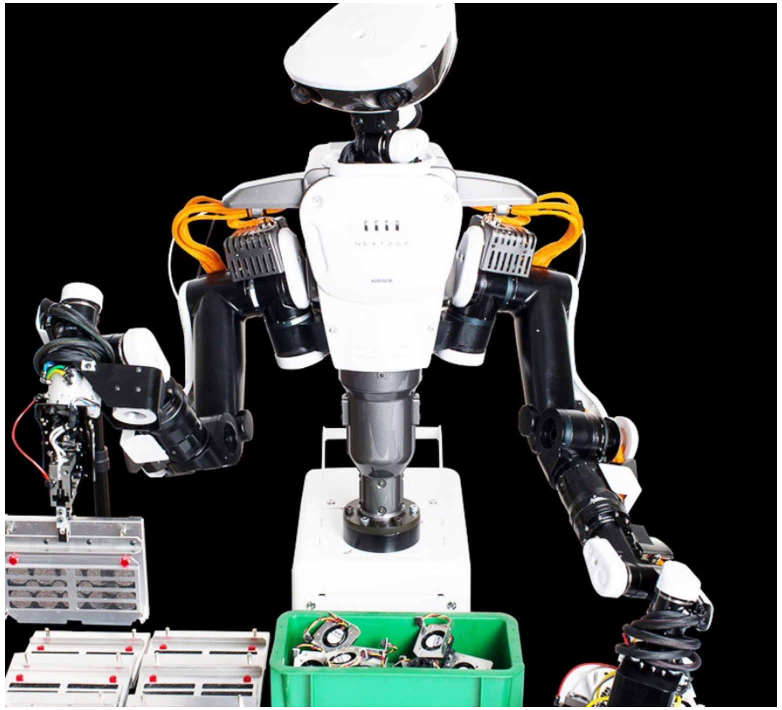
Image data from Mounted Camera $\xrightarrow{\text{CAE}}$ Extracted image feature

Current joint angles

Concatenated data $\xrightarrow{\text{MTRNN}}$ Predicted joint angle → Movement of robot

Experimental Setup

experimental evaluation setting



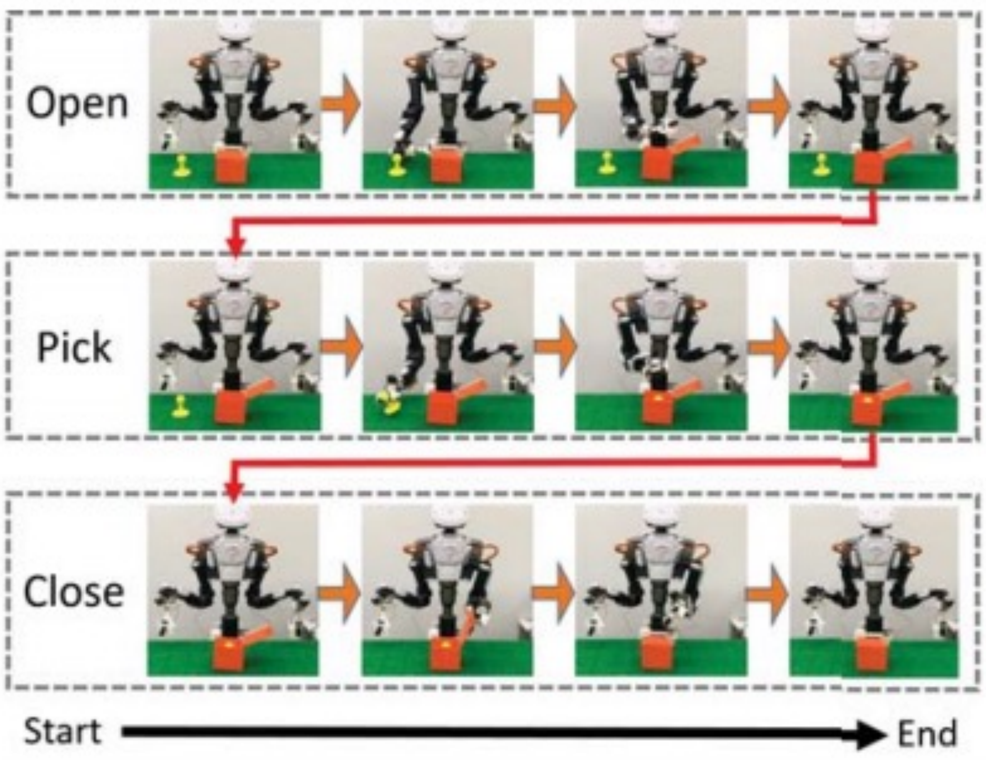
Nextage Open Robot from Kawada Robotics

Nextage *is a* ***humanoid robot*** *with a head-mounted camera and two arms, each Each arm has* ***six degrees of freedom*** *(DoF) and an additional gripper.*

AncoraSIR.com

Experimental Setup

experimental evaluation setting



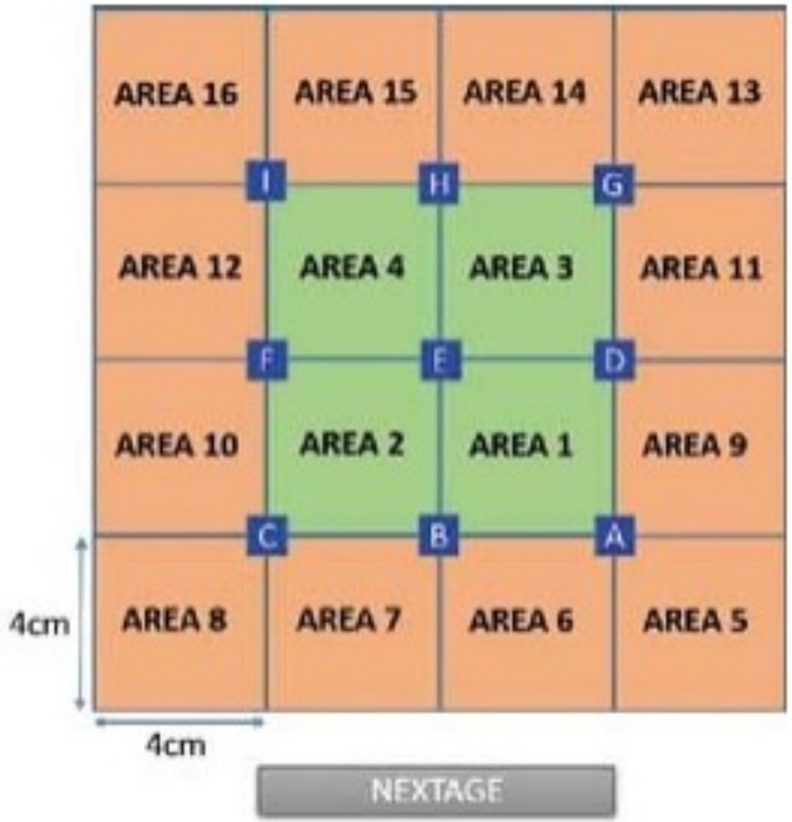
(1) Open a box
(2) Pick up an object and put it in the box
(3) Close the box

By combining these three subtasks, the robot performs a complete "**put into the box**" task. Each subtask is designed so that the **robot's initial and final positions of the robot are the same**, resulting in a **smoother transition between each subtask**. smooth transition between each subtask. In addition, since **the initial and final positions** of the robot and the MTRNN's initial and final internal states are the same, the only factor that causes the transition between subtasks is the image data. **The only factor that causes the transition between subtasks is the difference in image data**. Each subtask is **trained separately and there is no relationship between the three subtasks.**

AncoraSIR.com

Experimental Setup

experimental evaluation setting

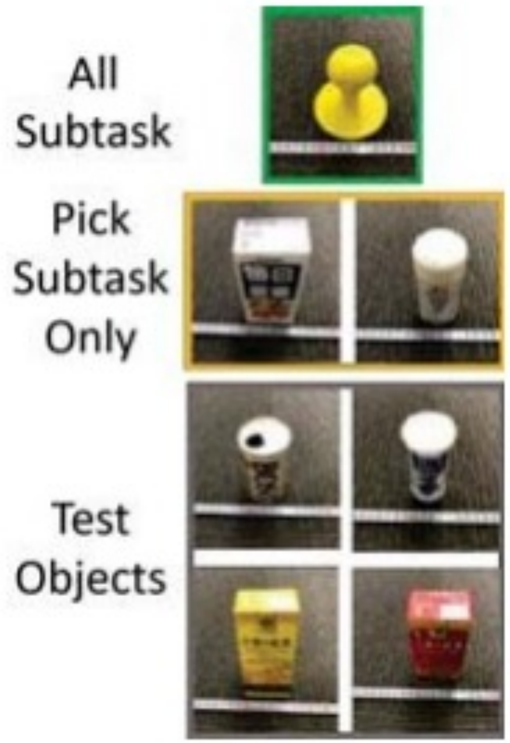


For the object manipulation, we divided the entire field into **16** squares (to facilitate robot positioning) and placed objects at each of the **9** intersections to complete the three operations described above, **and each subtask was trained three times at these different locations.**

AncoraSIR.com

Experimental Setup

experimental evaluation setting



In addition, in the picking subtask, the robot was also trained to trained on two additional objects: **a white pepper container** and **a white package of juice**. For the test group, we also prepared a **white steel can**, **a salt shaker**, **a yellow juice pouch and a red juice pouch** to test the generality relative to the picked objects.

AncoraSIR.com

Experimental Setup

experimental evaluation setting

DataSet

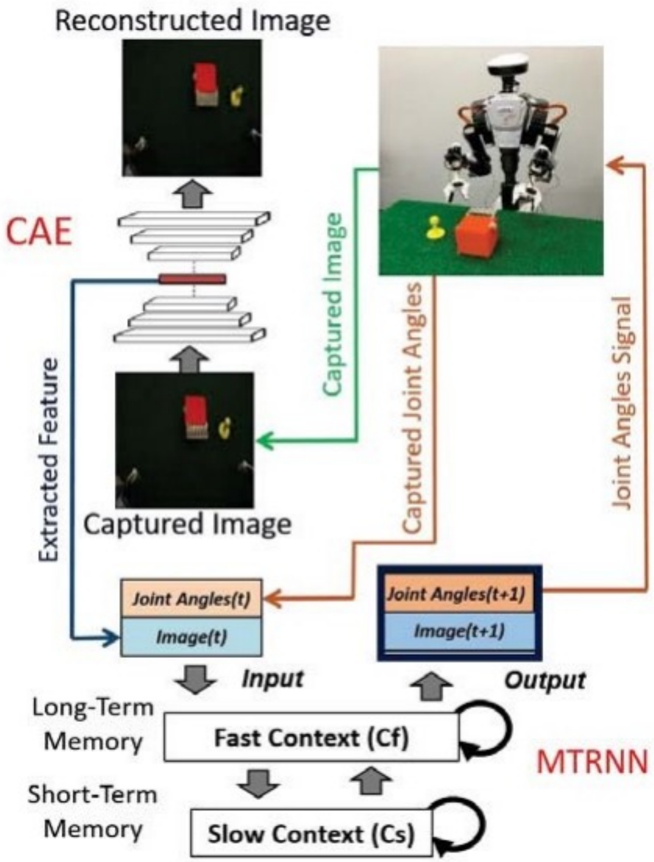
The training data for the robot was created directly by the experimenter using a 3D mousethat the motion and image data are sampled simultaneously.
The images and joint angles of thetraining data were sampled at a rate of 5 frames per second, and each subtask took approximately **78** seconds were required.
The robot's camera captured a **64x64** pixel RGB image with the number of **12,288**

AncoraSIR.com

Experimental Setup

experimental evaluation setting

DataSet



The training data was processed by CAE to **reduce its dimensionality to 20.**

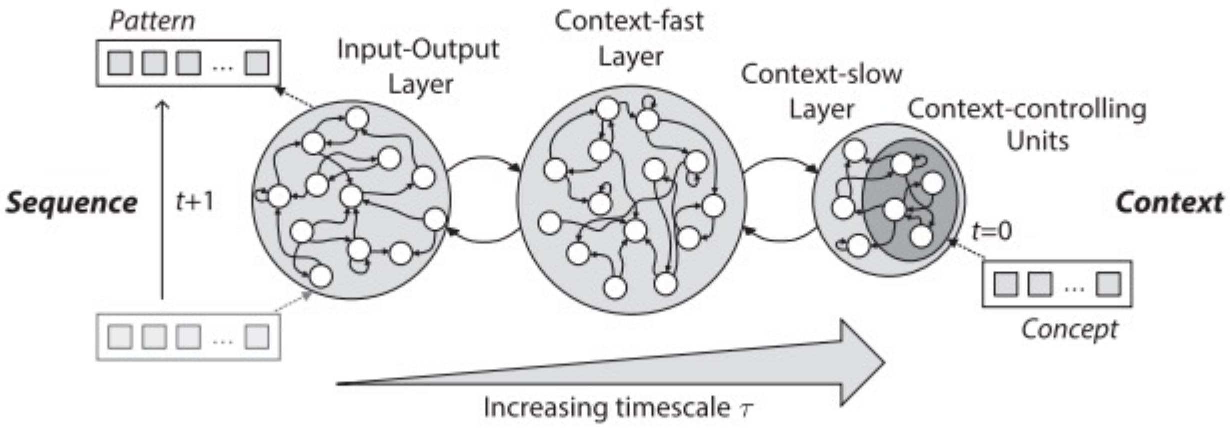
The TRNN training data are **20 raw image** features with the robot arm and gripper's **14 joint angles** connected to the data.

The training data for all subtasks **data were looped three times**, which means that the training data were prepared so that each sub-tasks were repeated three times to **stabilize and smooth the output.**

Experimental Setup

experimental evaluation setting

How to train



The overall MTRNN architecture with exemplary **three horizontally parallel layers: input-output(IO), context-fast(Cf), and context-slow (Cs).**
with increasing timescale t,where the Cs layer includes some context-controlling (Csc)units. While the IO layer processes dynamic patterns over time, the Csc units at first time step (t=0)contain the context of the sequence, where a certain concept can trigger the generation of the sequence.

AncoraSIR.com

Experimental Setup

experimental evaluation setting

How to train

During learning the MTRNN can be trained with sequences, and self-organises the weights and also the internal state values of the Csc units. The overall method can be a variant of *backpropagation through time* (BPTT), sped up with appropriate measures based on the task characteristics.

For instance, if the MTRNN produces continuous activity (IO) we can modify the input activation with a prorated *teacher forcing* (TF) signal $\alpha \in]0,1[$ of the desired output y^* together with the generated output y of the last time step

$$x_{t,i} = \begin{cases} (\alpha) y^*_{t-1,i} + (1-\alpha) y_{t-1,i} & \text{iff } t \geq 1 \wedge i \in I_{IO}, \\ y_{t-1,i} & \text{iff } t \geq 1 \wedge i \notin I_{IO}. \end{cases}$$

SUSTech
Southern University of Science and Technology

Experimental Setup

experimental evaluation setting

How to train

In the forward pass, an appropriate error function E is accumulating the error between activation values (y) and desired activation values (y^*) of IO neurons at every time step based on the utilised activation function. In the second step, the partial derivatives of calculated activation (y) and desired activation (y^*) are derived in a backward pass. In the case of, e.g. a decisive normalisation function (softmax) in IO and a sigmoidal function f_{sig} in all other layers, we can specify the error on the internal states of all neurons as follows:

$$\frac{\partial E}{\partial z_{t,i}} = \begin{cases} y_{t,i} - y^*_{t,i} + \left(1 - \frac{1}{\tau_i}\right) \frac{\partial E}{\partial z_{t+1,i}} & \text{iff } i \in I_{\text{IO}}, \\ \sum_{k \in I_{\text{All}}} \frac{w_{ki}}{\tau_k} \frac{\partial E}{\partial z_{t+1,k}} f'_{\text{sig}}(z_{t,i}) + \left(1 - \frac{1}{\tau_i}\right) \frac{\partial E}{\partial z_{t+1,i}} & \text{otherwise,} \end{cases}$$

AncoraSIR.com

Experimental Setup

experimental evaluation setting

How to train

where the gradients are 0 for the time step $T+1$. For the error function E of the decisive normalisation the *Kullback–Leibler divergence* (KLD) is used, where the cross-entropy is generalised to $|I_{IO}|$ classes (Kullback & Leibler, 1951). Importantly, the error propagated back from future time steps is particularly dependent on the (different) timescales.

Finally, in every epoch n the weights w but also biases b are updated

$$w_{ij}^{(n)} = w_{ij}^{(n-1)} - \eta_{ij} \frac{\partial E}{\partial w_{ij}} = w_{ij} - \eta_{ij} \sum_t \frac{1}{\tau_i} \frac{\partial E}{\partial z_{t,i}} x_{t,j},$$

$$b_i^{(n)} = b_i^{(n-1)} - \beta_i \frac{\partial E}{\partial b_i} = b_i - \beta_i \sum_t \frac{1}{\tau_i} \frac{\partial E}{\partial z_{t,i}},$$

AncoraSIR.com

Experimental Setup

experimental evaluation setting

How to train

where the partial derivatives for w and b are, respectively, the accumulated sums of weight and bias changes over the whole sequence, and η and β denote the learning rates for weight and bias changes. To facilitate the application of different methods for speeding up training, we can use individual learning rates for all weights and biases to allow for individual modifications of the weight and bias updates, respectively.

The initial internal states $c_{0,i}$ of the Csc units define the behaviour of the network and are also updated as follows:

$$c_{0,i}^{(n)} = c_{0,i}^{(n-1)} - \zeta_i \frac{\partial E}{\partial c_{0,i}} = c_{0,i} - \zeta_i \frac{1}{\tau_i} \frac{\partial E}{\partial z_{0,i}} \quad \text{iff } i \in I_{Csc},$$

where ζ_i denotes the learning rates for the initial internal state changes.

AncoraSIR.com

Experimental Setup

Network Parameters

experimental evaluation setting

The CAE was trained over 1,500,000 iterations with a learning rate of $\alpha = 0.0002$, $\beta_1 = 0.90$, $\beta_2 = 0.999$, and $\varepsilon = 10^{-8}$. Here, the mini-batch size was 100. The MTRNN was trained over 150,000 epochs with a learning rate of $\alpha = 0.001$, $\beta_1 = 0.90$, $\beta_2 = 0.999$, and $\varepsilon = 10^{-8}$. Here, the weight decay was 10^{-3}. Both the CAE and the MTRNN are optimized with the Adam. Other details about the CAE and MTRNN used in this study are shown in Table I.

TABLE I: Neural Network Parameters

DCNN	conv@3ch - conv@32ch - conv@64ch - conv@128ch - conv@256ch - full@1000 - full@20 - full@1000 - dconv@256ch - dconv@128ch - dconv@64ch - dconv@32ch - dconv@3ch
MTRNN	Input_nodes@34 - Cf_nodes@100 - Cs_nodes@5 Cf_τ@5 , Cs_τ@70

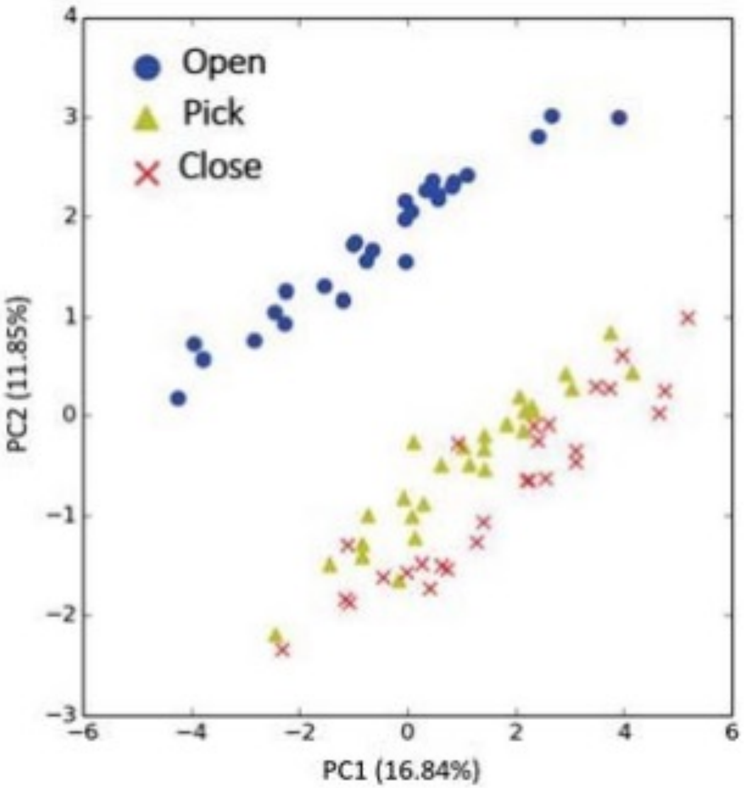
conv: convolutional layer dconv: deconvolutional layer
full: fully-connected layer

AncoraSIR.com

Experimental Results

Further explaination of the title with supporting evidence

Extraction of image features



From the data:

It is clear that the images of the first and second tasks are clearly separated, while the second and third tasks are aggregated together.

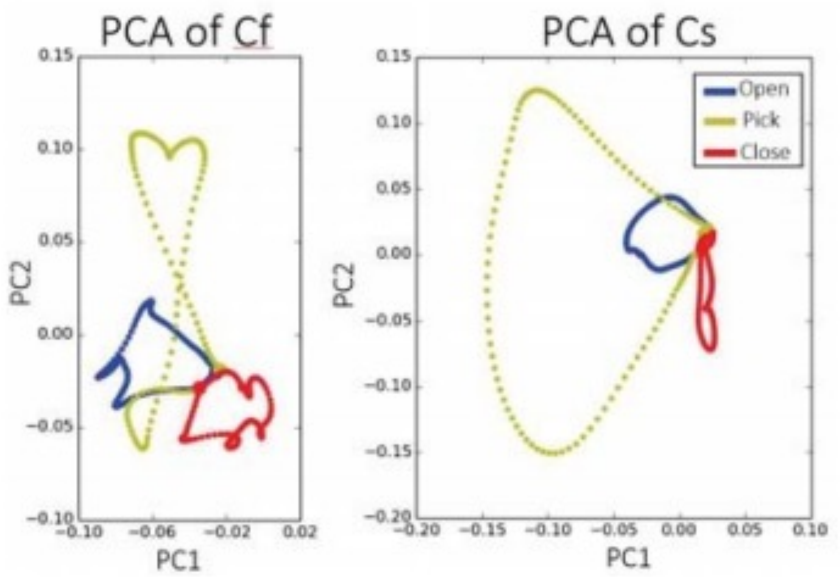
The reason for this problem may be that the object we are operating on is too small compared to the box we are placing, so the pickup and close box operations do not have a significant change on the overall image

AncoraSIR.com

Experimental Results

Further explaination of the title with supporting evidence

Generate a complete set of complex movements



The results show that **the contextual layers** start and end with similar values, thus forming attraction points. Each subtask starts from the **strong attraction** point generated by the constraints **of MTRNN attraction point and diverges from that point using the input difference of MTRNN.**

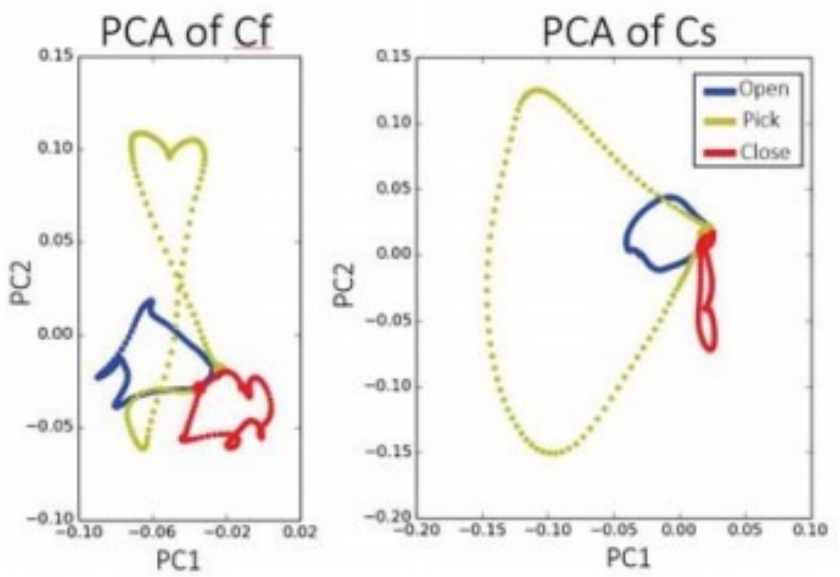
Since the input difference at the beginning of each subtask is the image data, the data is used to **disperse from the attraction point,** generate motion corresponding to the image data, and converge to the attraction point.

AncoraSIR.com

Experimental Results

Further explaination of the title with supporting evidence

Generate a complete set of complex movements



Train patterns of multiple discrete actions with constraints that allow the robot to able to **perform actions that are difficult when trained continuously**. One advantage of this is the achievement of repetition capability, allowing the robot to **repeat the same task** in a shorter time period **performing the same task repeatedly.**

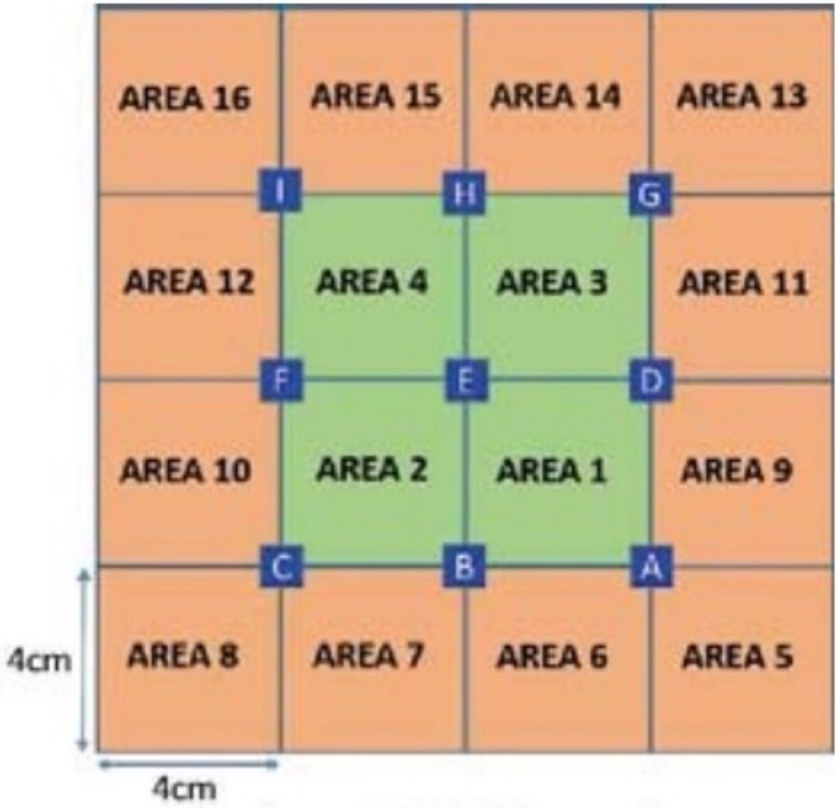
AncoraSIR.com

Discussion of Results

Further explaination of the title with supporting evidence

Task-specific experimental results

In the area shown in the figure, the robot performing the pick-up subtask was tested. Each area was tested five times with yellow stamped objects. The robot could pick up objects in **zones 1 to 4** perfectly. The robot was also able to pick up objects in **zones 5 to 16,** but failed in the attempt. The total assembly power for picking up objects in 16 zones power was 88.75% (80 attempts).



AncoraSIR.com

Discussion of Results

Further explaination of the title with supporting evidence

Generalization of tasks

CAE trained with only yellow stamps **cannot clearly reconstruct** other objects' images, **which affects the accuracy of robot motion generation**. To confirm the CAE effectiveness, we tested the robot generation with all objects to pick subtasks.

The ability of the robot to generate the picking subtask was tested with all objects. For the red juice package, the robot could not identify when it needed to perform the pick subtask. Here, it generated the close subtask. For the other objects, the robot generated the picking subtask, but failed to pick the untrained objects **(it knocked them over with its paws).**

Discussion of Results

Further explaination of the title with supporting evidence

Generalization of tasks

CAEs trained with other objects compared to those trained with only the yellow stamp object.
CAE can reconstruct **other untrained objects more clearly and vividly than objects**. With the CAE trained with additional objects, the robot **generated all objects for the pickup subtasks** and was able **to pick up objects without knocking them over.**

AncoraSIR.com

SUSTech
Southern University of Science and Technology

Discussion of Results

Further explaination of the title with supporting evidence

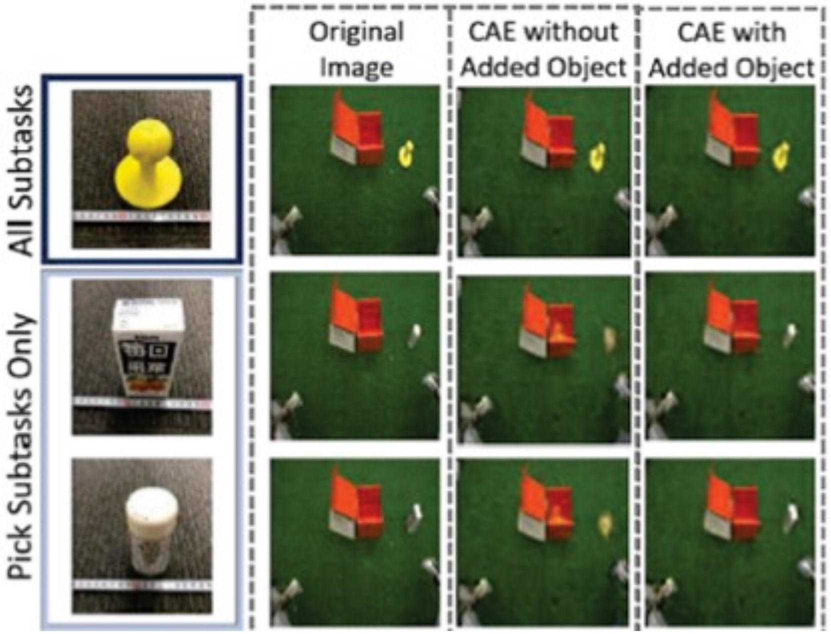
Generalization of tasks

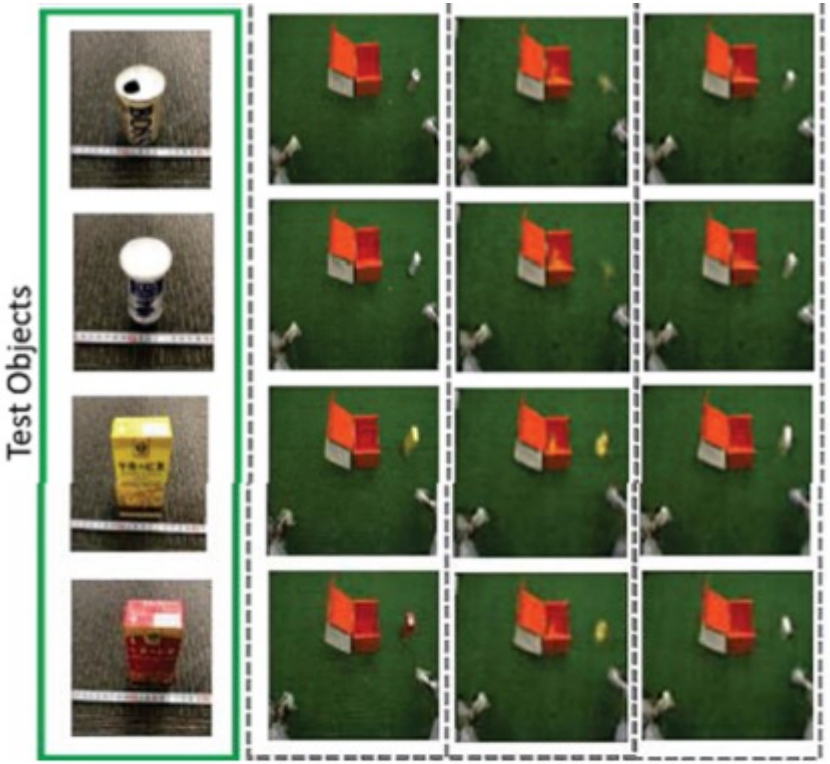
Although the robot was **not trained to generate open** or close subtasks with these objects, the robot succeeded in **generating a complete boxing task for other objects trained with only the picking subtask**. The complete boxing task was generated for other objects trained with only the picking subtask.
In addition, **with the exception of the red juice bag**, the robot could generate a complete boxing task for all untrained objects. robot can generate complete tasks for all untrained objects except the **red juice bag**. For this object, the robot successfully generated the opening and picking subtasks, **but failed to convert them to but failed to convert to the closing subtask and repeated the picking subtask.**

Discussion of Results

Further explaination of the title with supporting evidence

Generalization of tasks





Training with additional objects CAE reconstructed the original image more accurately than other CAEs.

AncoraSIR.com

Critique

Based on

“a robot demonstration of a human operator using a 3d mouse controller”,

predictive learning models and deep neural networks (RNN)

enable it to perform

multiple short tasks and autonomously combine them

according to the situation to

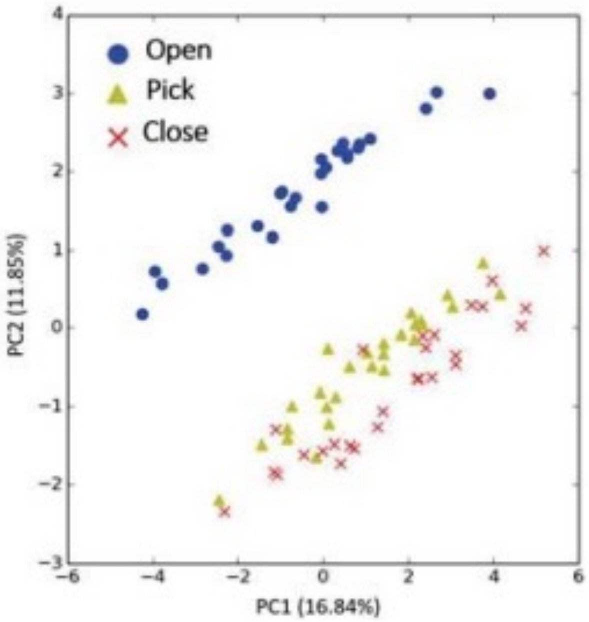
generate composite tasks

AncoraSIR.com

SUSTech
Southern University of Science and Technology

Critique

1. Previous research is clear,
   learning and classifying previous research results
2. The selection of methods is appropriate,
   and the advantages and limitations of each method are summarized
3. Good pointcut,
   long complex tasks, separate training, highly constructive
4. The evaluation experiment design is quite corresponding,
   “open pick close”

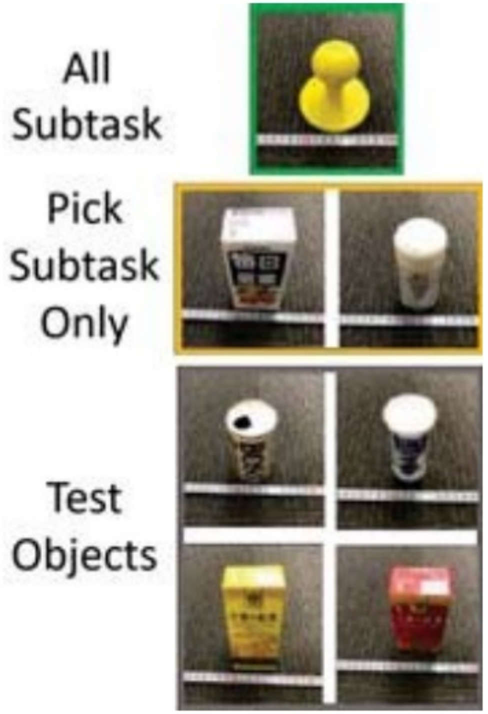


AncoraSIR.com

Limitations

There is something to be thought about in the practical application of the project

1. The connection of short tasks is relatively simple (there is no problem that items are not easy to catch or throw, which disrupts the process).
2. Constraint strength and conditions can be increased in constrained/unconstrained MTRNN to show more
3. For identifying objects and grasping paths, the data of previous studies can be used for reference to improve efficiency

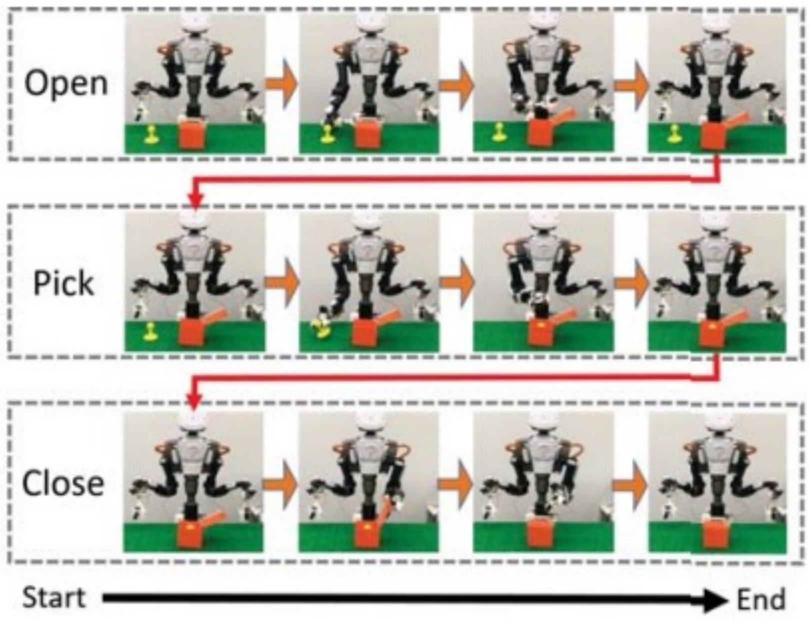


AncoraSIR.com

Future Work for Paper / Reading

A task robot capable of handling dynamic information and interference

1. Make task more difficult, and increase the number.
2. Study other advanced studies on short tasks, optimize the learning process and results.
3. Change the relevance and connection, and further discuss the switching ability of this method among subtasks
4. Let it handle series of different long tasks autoly with enough training sets of short tasks
5. Optimize the process by algorithm, repeatability and sequencing,

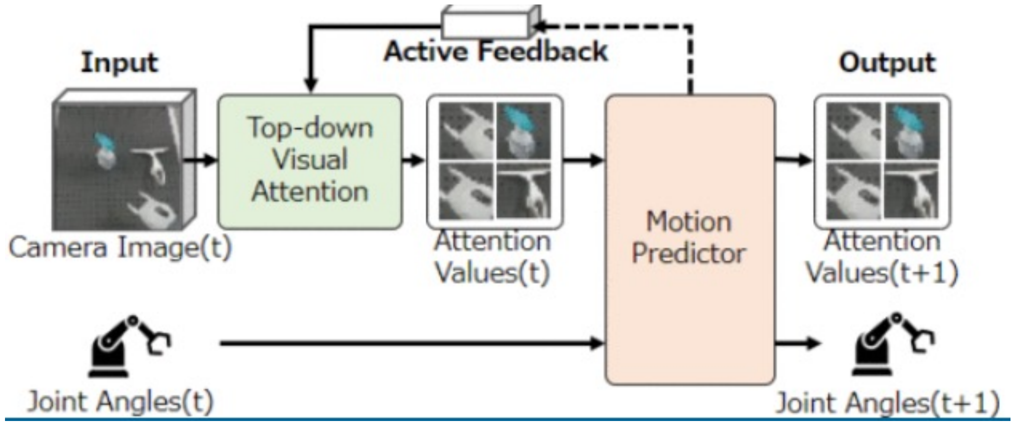


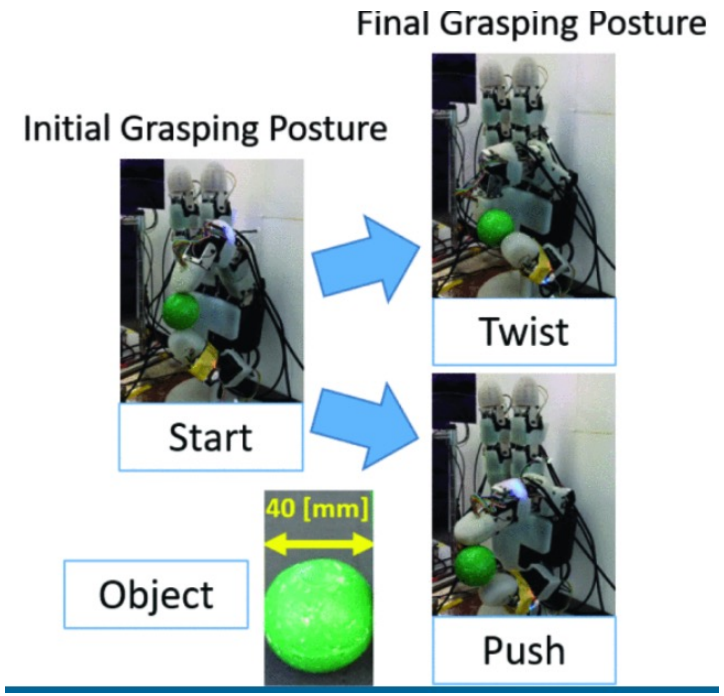
AncoraSIR.com

Future Work for Paper / Reading

A task robot capable of handling dynamic information and interference

1. Autopilot compressed images
2. Towel folding robot with six degrees of freedom
3. Improve the stability of cooperative robots based on human vision
4. Robot hand manipulation "Twist" and "Push"





AncoraSIR.com

Extended Readings

Further explaination of the title with supporting evidence

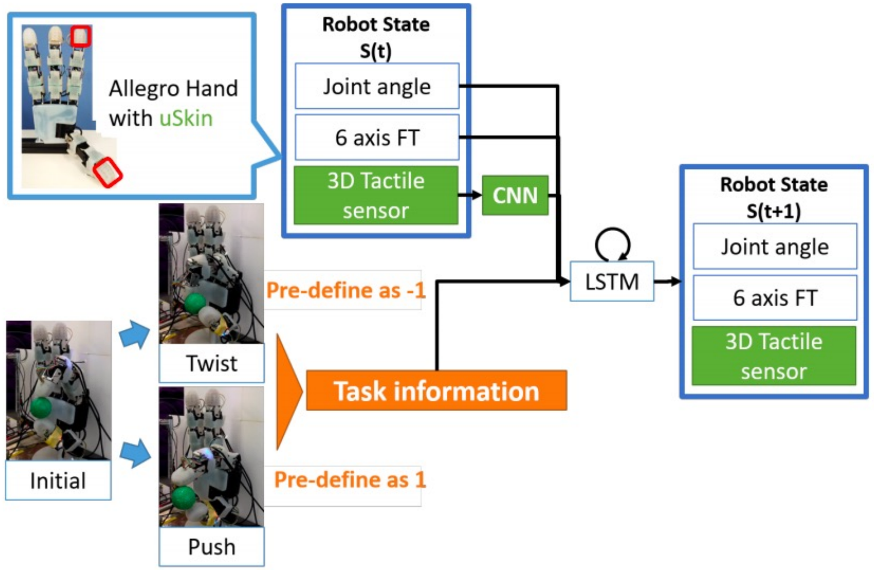
Variable In-Hand Manipulations for Tactile-Driven Robot Hand via CNN-LSTM

in 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) October 25-29, 2020, Las Vegas, NV, USA (Virtual)

～ In this article, the author also used Predictive learning, which has a very different purpose and effect than the articles listed here.

～ In this article, the purpose of training is to be able to performing various in-hand manipulation tasks, without learning each individual task.

～ In the current paper we use two fingers of the Allegro hand, and each fingertip is anthropomorphically shaped and equipped not only with 6-axis force-torque (F/T) sensors, but also with uSkin tactile sensors, which provide 24 tri-axial measurements per fingertip. A convolutional neural network is used to process the high dimensional uSkin information, and a long short_x0002_term memory (LSTM) handles the time-series information. The network is trained to generate two different motions ("twist" and "push"). The desired motion is provided as a task_x0002_parameter to the network, with twist defined as -1 and push as +1. When values between -1 and +1 are used as the task parameter, the network is able to generate untrained motions in-between the two trained motions.



AncoraSIR.com

Summary

Further explaination of the title with supporting evidence

Problem:

Subtask design, Switching among multiple tasks，The design of neural network

Importance：

Adapt to various working environment, generate and combine various tasks

Key limitation of prior work：

Few studies have addressed switching among mutiple tasks

Key insights of the proposed work:

propose a model that uses two DNNs to generate mul_x0002_tiple short discrete subtasks to complete a longer target task.

What did they demonstrate by this insight:

With our model, the robot switches among short subtasks autonomously by recognizing images. Then, the robot executes the target task.

AncoraSIR.com

Reference

- [1] G. Hinton and R. R. Salakhutdinov, "Science," Science, vol. 313, no. July, pp. 504–507, 2006.
- [2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," Advances in Neural Information Processing Systems 25, pp. 1097–1105, 2012.
- [3] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-End Training of Deep Visuomotor Policies," ArXiv:1504.00702, 2015.
- [4] W. Han, S. Levine, and P. Abbeel, "Learning compound multi-step controllers under unknown dynamics," IEEE International Conference on Intelligent Robots and Systems, vol. 2015-December, pp. 6435– 6442, 2015.
- [5] Y. Yamashita and J. Tani, "Emergence of functional hierarchy in a multiple timescale neural network model: A humanoid robot experiment," PLoS Computational Biology, vol. 4, no. 11, 2008. [16] K. Noda, H. Arie, Y. Suga, and T. Ogata, "Multimodal integration learning of robot behavior using deep neural networks," Robotics and Autonomous Systems, vol. 62, no. 6, pp. 721–736, 2014.
- [6] K. Noda, H. Arie, Y. Suga, and T. Ogata, "Multimodal integration learning of robot behavior using deep neural networks," Robotics and Autonomous Systems, vol. 62, no. 6, pp. 721–736, 2014.
- [7] R. Rahmatizadeh, P. Abolghasemi, L. Boloni, and S. Levine, "VisionBased Multi-Task Manipulation for Inexpensive Robots Using EndTo-End Learning from Demonstration," arXiv:1707.02920, 2017.

SUSTech Design + Learning Lab PowerPoint Template

Name

Affiliation

Supervisor

Contact

仿生设计与学习实验室 Bionic Design & Learning Lab

main DL In Collaboration with MIT

AncoraSIR.com

SUSTech Southern University of Science and Technology