Natural Teaching of Robot-Assisted Rearranging Exercises for Cognitive Training *

Antonio Andriella, Alejandro Suárez-Hernández, Javier Segovia-Aguas, Carme Torras, and Guillem Alenyà

Institut de Robòtica i Informàtica Industrial, CSIC-UPC,
C/Llorens i Artigas 4-6, 08028 Barcelona.
{aandriella, asuarez, jsegovia, torras, galenya}@iri.upc.edu

Abstract. Social Assistive Robots are a powerful tool to be used in patients' cognitive training. The purpose of this study is to evaluate a new methodology to enable caregivers to teach cognitive exercises to the robot in an easy and natural way. We build upon our existing framework, in which a robot is employed to provide encouragement and hints while a patient is physically playing a cognitive exercise. In this paper, we focus on empowering the caregiver to easily teach new board exercises to the robot by providing positive examples.
The proposed learning method has two main advantages i) the teaching procedure is human-friendly ii) the produced exercise rules are human-understandable. The learning algorithm is validated in 6 exercises with different characteristics, correctly identifying and representing the rules from a few examples.

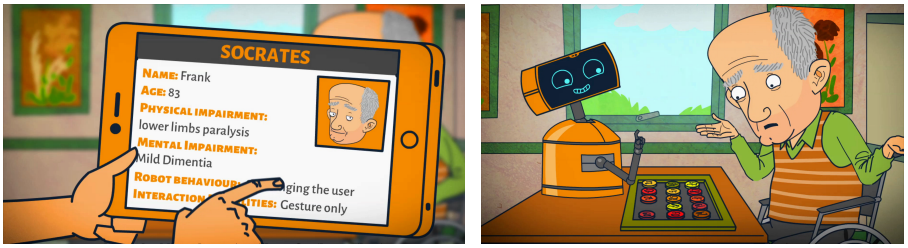
Keywords: SAR · Robotic Assisted Exercises · Natural Teaching.

1 Introduction

The increase in life expectancy is one of the most important achievements of the 21st century. However, ageing and age-related diseases are a mounting challenge for families, social, economic and healthcare systems [15]. One of the biggest challenges of the modern world associated with the ageing population is dementia [13]. According to the Word Health Organization the number of people with dementia will rise from 50 mil in 2018 to 82 mil in 2030, and more than 150 mil in 2050 [1].

Currently there is no treatment available to cure dementia or to modify the progression of the disease [7]. The limited efficacy of the pharmacological therapies is the reason to explain the arising interest for non-pharmacological interventions for dementia patients. The non-pharmacologic intervention aims to enhance or at least maintain the individuals cognitive function, enabling to

* This project has received funding from the European Union´s Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement SOCRATES MSCA-ITN-721619, by the Spanish Ministry of Science and Innovation HuMoUR TIN2017-90086-R, and by the Spanish State Research Agency through the María de Maeztu Seal of Excellence to IRI (MDM-2016-0656).



(a) First loop of interaction (b) Second loop of interaction

Fig. 1. Illustrative images. (a) A caregiver sets the patient's mental and physical impairment and the robot's initial behaviour. (b) The robot assists the patient while he is playing the cognitive exercise. Frames from the video in `https://youtu.be/zqcLSdl0UcE`.

address behavioural symptoms that often exist in people affected by dementia, such as depression [7]. These interventions can be divided into four categories outlined by [9]: holistic techniques, brief psychotherapy, cognitive methods, and alternative methods. In our project, we focus on cognitive therapies, specifically on Cognitive Training (CT) exercises. CT is one of those activities that seeks, through repetitive practice, to train specific cognitive processes via standardized exercises [6].

Socially Assistive Robotics (SAR) is a branch of Robotics that aims to endow robots with the capability to aid people through individual social assistance, rather than physical, in convalescence, rehabilitation, and training [16]. Robots not only can be available twenty-four hours a day, but they may also help with the growing shortage of personnel support and, moreover, ease the workload of human therapists. Research has already shown that SAR can help improve the quality of life for older adults and bridge the gap when human assistance is not available [20].

In this paper, we extend our previous work [4] in which a SAR is employed by a caregiver to provide encouragement and motivation, through speech and gesture, to a patient while he is playing a cognitive exercise. There, two main loops of interaction are proposed. In the first one, the caregiver sets the patient's cognitive and physical impairment and the initial preferences on the robot's behaviour (see Fig. 1a). In the second one, the robot, given the caregiver's settings, provides assistance to the user through encouragements and hints based on his performance (See Fig. 1b).

Most of the current works in SAR are focused on the second loop of interaction (robot - final user) in which the robot, partially or entirely replaces the caregiver's role [2]. On the contrary, we target our attention to the first loop as we envisage a central role for the caregiver.

In previous work [4], the caregiver could provide information about the patient with the intent to personalize the robot's behaviour (see Figure 1a). However, she/he could not extend the repertoire of exercises. Programming a robot, in fact, is a tedious process that requires a considerable amount of technical ex-

pertise. With that in mind, the practice of configuring new exercises is exclusive to the competent personnel and this represent a limitation in reality.

In this paper, we present a way to equip the caregiver with an easy and natural human-like approach to teach new exercises simply by playing them. From few moves taught by the caregiver, the robot is able to learn the rules of the new exercise. This enables the robot to monitor patients who perform this same exercise. Furthermore, the learned rules are available in first-order language. This provides an explainable and intuitive way of understanding what the system has learned, since rules can be easily translated to natural language.

The proposed exercises are inspired by the Syndrom Kurztest [17], and they have been thought specifically to combine cognitive and motor functions on visuomotor skills like grasping and manipulation [11]. Six different exercises are defined: sorting odd numbers in ascending order, sorting numbers in ascending and descending order, an exercise where position within the board matters, composing a word, and sorting letters in alphabetical ascending order.

The main contributions of this paper are: (1) a friendly method of teaching board exercises using natural interactions; and (2) a learning algorithm that produces human-understandable rules.

We believe the proposed paper is a step further in the direction to provide non-expert people, and in particular therapists, with easy-to-use methods where exercises can be programmed through playing examples, and directly represent exercise rules into an explainable symbolic high-level language such as STRIPS [12].

2 Related Work

In the last decade, a lot of effort has been put in developing robotic systems programmable from non-experts.

Graphical programming and user-friendly interface have been developed to provide non-experts with a way to understand and program without investing much time in learning. Pieska *et al.* [18] present an interesting review about the state of the art on user-friendly interface for robots and in their work they develop a platform suitable for both programming experts and people with no robot programming skill. The platform is based on a set of ready-made plugins, whose plugins can be connected through a graphical interface to generate a robot program.

There are also a few attempts to introduce user-friendly programming to Robotic Operating System (ROS)-based systems. Crick *et al.* [10] introduce rosbridge, a middleware abstraction layer to enable ROS accessible to programmers that are not them-self roboticists. Tiddi *et al.* [23] develop a user interface for assisting non-expert users to design complex robot behaviours and control robotic systems based on ROS. Zubrycky *et al.* [24] present a graphical programming interface called Robokol, based on ROS and Snap, that enables non-technical professionals to program robots and internet-of-things devices.

In the field of education robots, there are also examples of introducing easy interfaces to program robots. One of the most well-known is *Choregraphe*, a

(a) (b) (c)

Fig. 2. (a) Empty Board: goal row in white, storage rows in blue. (b) Tokens randomly placed in the storage rows from the caregiver. (c) Tokens moved in the goal row in ascending order from the caregiver.

graphical environment developed by Aldebaran Robotics for programming their humanoid robot, NAO [19]. *Choregraphe* enables non-experts to create robot behaviours using a Box library. It contains from high-level (walking, dancing, etc) to low level robot's functionalities (sensors, LEDs, etc) that can be assembled and linked based on user's own need to create a custom robot's behaviour. All these approaches have in common the idea of a graphical interface to link basic behaviours. On the contrary, we propose to use simple real demonstrations.

Real demonstrations have been used before to teach physical tasks using Learning by Demonstration techniques [5]. This approach has been extensively used in the literature for learning low-level robot motions [8], and also for incrementally learning assembly tasks [22]. In our work, we are not interested in teaching the motion trajectories but the logical rules of the exercise from a few user's gameplay demonstrations.

3 Method

A practical system ought to be naturally programmable by caregivers and practitioners. Thus, we propose that the caregiver shows the robot some successful runs of the exercise, and then let the robot discover the underlying rules. The board exercise is of the form of several tokens arranged (see Fig. 2b) in the **storage rows** (see blue cells in Fig. 2a) that have to be displaced to the **goal row** (see white cells of Fig. 2a) based on rules taught by the therapist (see Fig. 2c). This setup is very flexible, as the tokens can be labelled with numbers, letters or symbols, allowing the generation of multiple exercises. In this paper, we have used 6 different exercises to show the generality of the approach. We will present them in Sec. 4.

3.1 Perception

The most common approach is to use computer vision to monitor the changes in the environment [14]. However, from our experience in real environments, simple vision algorithms able to run embedded in common robots are not reliable enough

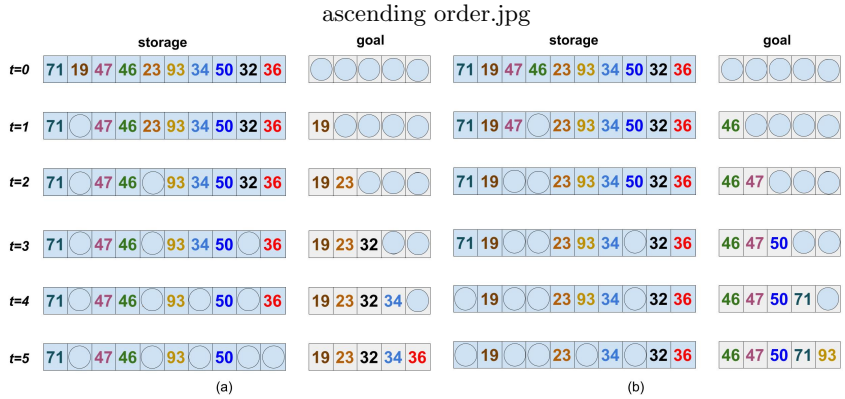


Fig. 3. Example of two traces or play-outs, provided by a caregiver to teach the "ascending order" exercise. Both traces are meant to demonstrate the same exercise. In (a) the caregiver teaches how to sort the five smallest numbers while in (b) he teaches how to sort the five largest numbers. In both traces, the initial token position ($t = 0$) is random. Later steps ($t > 0$) show the progress of the caregiver in the current trace.

in natural environments. Furthermore, most of the current robotic platforms are equipped with only one camera that is generally located in their head, thus if the camera is used to detect the board state then, for instance, it cannot be used to monitor the user's affective behaviour.

For this reason, we built an electronic board based on RFID technology . Each token is univocally identified with an ID. The board consists of 20 Grove - NFC boards one for each board cell, 3 Adafruit TCA9548A, 1 to 8 I2C multiplexers, and 1 Arduino Nano. The Grove NFC has a highly integrated transceiver module PN532 which handles contact-less communication at 13.56MHz. Finally, Grove NFC is controlled using I2C communication protocol.

Using this board we have experienced no errors in detecting the tokens and in creating reliable traces. A trace is a sequence of valid pick-and-place actions in which the demonstrator moves tokens from one location to another one (see Fig. 3). For each timestamp t, our driver records the current move as a triplet $(token\_id, orig, dest)$. It is worth to be mentioned that although the board is capable to detect more than one move at the same time, in this context we don't allow the demonstrator to teach more than one move at a time.

3.2 Actions and Rules

In every board scenario, all 10 tokens are distributed among the storage and goal cells. The particular arrangement conforms the current *state* of the exercise. The state is described in terms of propositional variables (e.g. when contains (l_{ij}, A) is true, it means that token A is in the cell (i, j)). An action is defined as a 2-tuple $\langle precondition, effect \rangle$, where *precondition* is a logic formula that must hold in the current state for the action to be applicable, and *effect* is a set of variable assignments that modify the current state. The exercise's *rules* determine the

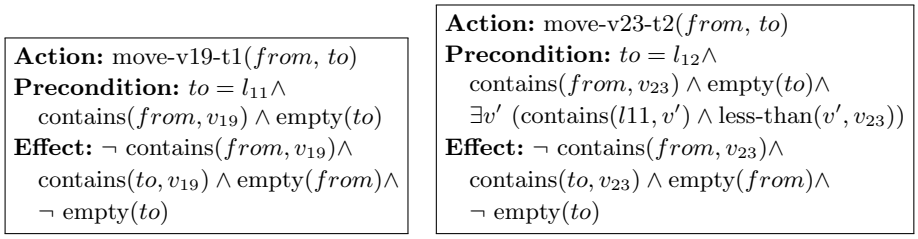


Fig. 4. (a) the action for moving token v_{19} (the token with value 19) as the first move requires that the destination is the top left corner (location l_{11}); (b) the action for allocating token v_{23} requires that the destination is at location l_{12} (at the right of the top left corner) and that there is a token at location l_{11} with lower numeric value.

set of applicable actions for each state, while the inapplicable actions are those against the rules. Therefore, learning the rules of an exercise reduces to learn the preconditions of the different actions.

Rules are expressed in terms of observable features that describe the characteristics of individual tokens (e.g. *odd*) or pairs of tokens (e.g. *less-than*). These features aim at being generic enough to provide the capability of defining interesting cognitive exercises. For a particular exercise, the rules involve only a subset of these features, while the rest are distractors. Since exercises require only 5 movements, it is possible that up to 5 tokens are irrelevant, or that exercises have more than one correct solution (e.g. sorting 5 tokens out of 10 in ascending order). The fact that tokens can be randomly arranged in the initial state adds extra complexity, so learning the rules must generalize over any input order.

Once rules are learned, they can be used for validating actions performed by the patients. In validation, the robot checks if the preconditions of the applied action are met in the previous state. If not, the patient is informed the action performed is invalid and the movement is undone by the robot.

As a means to give intuition about actions and rules, let us consider the board setting from Fig. 2b and the exercise of sorting tokens with numbers in ascending order. Fig. 4 shows 2 out of the 30 actions that are needed to solve the exercise. Notice how the preconditions are quite different from each other and are in direct correspondence with the rules of the exercise. The first movement must be moving one of the 6 tokens (19-23-32-34-36-46) with lowest value to the top left corner cell. Further movements must fill successively the rest of the goal row with tokens making sure that the cells at their left have lower numeric values. Our next section outlines how our method is capable of learning such rules from exercise traces.

3.3 Learning Method

Our learning method infers the precondition of all actions from exercise traces given by the caregiver (Fig. 5). An action has the form move-*value-timestep*

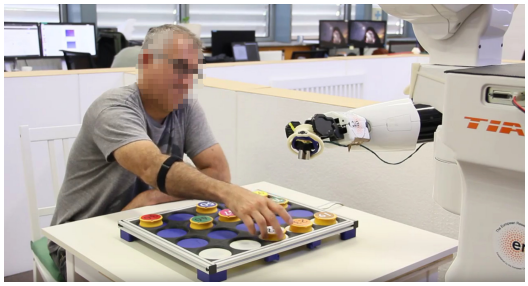


Fig. 5. A user teaching a new exercise to the robot.

(*from*, *to*), where *value* is the value of the token associated to the action, *timestep* is the move index in the sequence of moves in which the action is meant to be executed, *from* is the source cell parameter and *to* is the destination parameter. Since every exercise has 10 tokens and solutions require 5 time steps, problems will consist of learning preconditions of up to 50 pick-and-place actions.

The input is a set of traces or examples provided by the caregiver. Initially, we include all possible restrictions in all of the 50 actions' preconditions. Therefore, actions are initially inapplicable because some propositional variables interfere (e.g. the preconditions require that a token is in multiple locations at the same time). Our method then uses the available traces to progressively relax the preconditions, leaving them just restrictive enough so they can be used to explain the transitions of all the given traces. This is done via symbolic planning with Madagascar [21], in a way that is reminiscent of the approach taken and detailed by Aineto *et al.* [3]. Namely, we pose a planning problem in which the goal is to relax the preconditions in order to validate all the given traces.

Exercises like the *odd ascending* (introduced later in Sec. 4) do not need to use the whole set of actions, because many actions cannot ever be executed (like those actions for moving tokens with even numbers) and will not show up in any trace. Other exercises, like placing any sequence of 5 tokens in ascending order with no parity constraints, have many solutions and, thus, require more traces to learn the rules. Overall, the lower the complexity, the easier it is to learn the rules.

Fig. 3 shows two example traces that can be used to learn the rules of an ascending sorting exercise. These two traces serve to discard some of the irrelevant features (e.g. the initial position of the tokens). When exposed to enough traces, our learner infers the rules depicted in Fig. 4.

4 Experiments

We set up our system to gather exercise traces and learn human-understandable rules from them using the board presented in Sec. 3.1. We used 6 self-described exercises as a proof of concept: (i) sorting five odd numbers in ascending order; (ii) sorting five numbers in ascending order; (iii) sorting five numbers in

descending order, (iv) moving first storage row to goal row, (v) spell "CURIE" word choosing 5 of 10 available letters, and (vi) sorting five letters in lexicographical order. Exercises from (i) to (iv) use tokens with integer values, while exercises (v) and (vi) used tokens with letters.

4.1 Trace generation for new exercises

In order to learn new exercises, we need to extract traces such as those from Fig. 3. When an exercise is being demonstrated via traces, we transform the information from perception into symbolic states. Transitions between states are the result of actions taken by the caregiver. The number of required traces to learn the ground truth of the exercise will depend on the intrinsic complexity of exercise and solution space.

The minimum number of traces required to demonstrate an exercise depends on its complexity. Exercises (i) and (v), only require **2 traces** where all tokens are in different initial locations, as only one solution is possible. Exercises (ii), (iii) and (vi) need **12 traces** that summarize all the possible ways of sorting 5 pieces out of 10. Finally, exercise (iv)'s rules only involve locations, so we have to observe all 10 tokens moving from every column in the first storage row to the goal row, and this can be done with **10 traces**. Experimentally, we found that only expert users are able to provide this minimum number of demonstrations. Interestingly, our system can use all the traces that are provided, even if they are not the most informative ones.

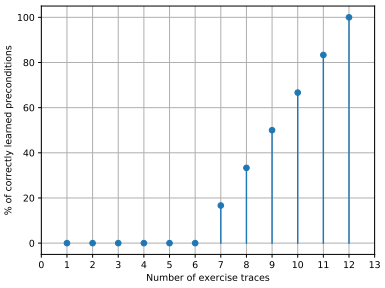
Additionally, our system provides an interface to decide beforehand which features are relevant for learning the next exercise (for example, an exercise may not require the *less-than* feature). When used, it reduces drastically the number of traces that are needed.

Fig. 6a gives an intuition on how the system's actions progress over time, as more and more traces are demonstrated. This plot is associated with the exercise (ii). When a few traces are shown, the system deduces overly restrictive rules. As more traces are shown, the system relaxes the preconditions and comes up with the correct set of rules.

4.2 Human Readable Rules

Figure 6b shows an example of the underlying rules behind an action in exercise (ii). We describe the method to generate human-understandable rules with a particular example: from the learned rules (Fig. 4b) an intermediate explanation is generated; using this intermediate representation, a complete sentence can be assembled. These rules can be displayed in natural language, so the caregiver can easily understand them.

Space restrictions do not allow to show all the generated rules. In terms of time, computing the complete set of rules and explanations for this kind of exercises take less than 2 seconds in an average computer.



(a)

Action: move-v23-t2($from$, to)
Inferred rules:
Destination (to) goal is cell l_{12}
Source ($from$) contains token v_{23}
Destination (to) is empty
Location l_{11} token is less than 23.
Descriptive text:
In the second movement, if the token marked with number 23 is in its original position then has to moved to the goal cell 2 only if it is empty and the token in cell 1 is less than 23.

(b)

Fig. 6. (a) percentage of correctly learned rules for exercise (ii) given the number of traces demonstrated by the caregiver (this is the percentage of actions whose precondition matches exactly the ground truth of the exercise); (b) rule from Fig. 4b (belonging to exercise (ii)) explained in natural language, as output by our system.

5 Conclusions and Future Work

The main contribution of this paper lies in proposing an approach to teach a robot new exercises through human demonstrations. The proposed method enables non-technical professionals, such as caregivers and therapists, to program new cognitive exercises to a robotic system, that can afterwards administer the exercise. Notably, the learned rules are easily readable and explainable because they can be expressed in logic language using well-understood features such as order relationships.

In the future, we will extend the current learning algorithm to cope with more complex rules. We would like to explore further its ability to generalize from examples, and reduce the number of traces needed to learn the rules.

Acknowledgements

Authors would like to thank Patrick Grosch, Sergi Hernandez and Alejandro López for assembling and programming the electronic board. Thanks to Nofar Sinai[1] for allowing us to use some frames of the SOCRATES video.

References

1. Alzheimer's disease facts and figures. Alzheimer's & Dementia pp. 367–429 (3 2018)
2. Abdi, J., et al.: Scoping review on the use of socially assistive robot technology in elderly care. BMJ Open **8**(2), 18815 (2018)
3. Aineto, D., Jiménez, S., Onaindia, E.: Learning STRIPS Action Models with Classical Planning. ICAPS (2018)

[1] http://www.vikkiacademy.com/

4. Andriella, A., et al.: Deciding the different robot roles for patient cognitive training. International Journal of Human Computer Studies **117**, 20–29 (3 2018)
5. Argall, B.D., et al.: A survey of robot learning from demonstration. Robotics and Autonomous Systems **57**(5), 469–483 (2009)
6. Bahar-Fuchs, A., et al.: Tailored and Adaptive Computerized Cognitive Training in Older Adults at Risk for Dementia: A Randomized Controlled Trial. Journal of Alzheimer's Disease **60**(3), 889–911 (2017)
7. Berg-Weger, M., Stewart, D.B.: Non-Pharmacologic Interventions for Persons with Dementia. Missouri medicine **114**(2), 116–119
8. Calinon, S.: Learning from Demonstration (Programming by Demonstration). In: Encyclopedia of Robotics, pp. 1–8 (2018)
9. Cammisuli, D.M., et al.: Non-pharmacological interventions for people with Alzheimer's Disease: A critical review of the scientific literature from the last ten years. European Geriatric Medicine **7**(1), 57–64 (2016)
10. Crick, C., et al.: Rosbridge: ROS for non-ROS users. In: Springer Tracts in Advanced Robotics. vol. 100, pp. 493–504 (2017)
11. De Boer, C., et al.: Thinking-While-Moving Exercises May Improve Cognition in Elderly with Mild Cognitive Deficits: A Proof-of-Principle Study. Dementia and Geriatric Cognitive Disorders Extra **8**(2), 248–258 (2018)
12. Fikes, R., Nilsson, N.: BLACKBOX: A New Approach to the Application of Theorem Proving to Problem Solving. Artificial Intelligence Journal **2**, 189–208 (1971)
13. Irwin, K., et al.: Healthy aging and dementia: Two roads diverging in midlife? (2018)
14. Leo, M., et al.: Computer vision for assistive technologies. Computer Vision and Image Understanding **154**, 1–15 (2017)
15. Lunenfeld, B., Stratton, P.: The clinical consequences of an ageing world and preventive strategies. Best Practice and Research: Clinical Obstetrics and Gynaecology **27**(5), 643–659 (10 2013)
16. Matarić, M.J., Scassellati, B.: Socially Assistive Robotics. Springer Handbook of Robotics **6**, 1973–1994 (2016)
17. Overall, J.E., Schaltenbrand, R.: The SKT neuropsychological test battery. Journal of Geriatric Psychiatry Neurology **5**(0891-9887), 220–227 (1992)
18. Pieskä, S., Pieskäa, J., Saukko, O.: Towards easier human-robot interaction to help inexperienced operators in SMEs. 3rd IEEE International Conference on CogInfoCom 2012 - Proceedings pp. 333–338 (2012)
19. Pot, E., et al.: Choregraphe: A graphical tool for humanoid robot programming. In: Proceedings - IEEE International Workshop on RO-MAN. pp. 46–51 (2009)
20. Pu, L., et al.: The Effectiveness of Social Robots for Older Adults: A Systematic Review and Meta-Analysis of Randomized Controlled Studies (2019)
21. Rintanen, J.: Madagascar: Scalable Planning with SAT. In: Proceedings of the 8th International Planning Competition (2014)
22. Savarimuthu, T.R., et al.: Teaching a Robot the Semantics of Assembly Tasks. IEEE Transactions on Systems, Man, and Cybernetics: Systems pp. 670–692 (2018)
23. Tiddi, I., et al.: A user-friendly interface to control ROS robotic platforms. CEUR Workshop Proceedings **2180** (2018)
24. Zubrycki, I., Kolesiński, M., Granosik, G.: Graphical programming interface for enabling non-technical professionals to program robots and internet-of-things devices. Advances in Computational Intelligence **10306**, 620–631 (2017)