Implicit Association Test

Introduction

Implicit-association test (IAT) is a classic psychological task intended to measure subconscious associations between mental bias of objects in memory. It is thought to measure implicit altitude, so it can show the individual's relative real thoughts, automatic response and avoid social-desirability bias. It especially can be useful to measure the gender, age discrimination or racism of people.

Here, I use IAT to measure the young students' implicit altitude towards 'old names' - if they view old names as outgroup and if they view young names as ingroup.

Experimental procedure

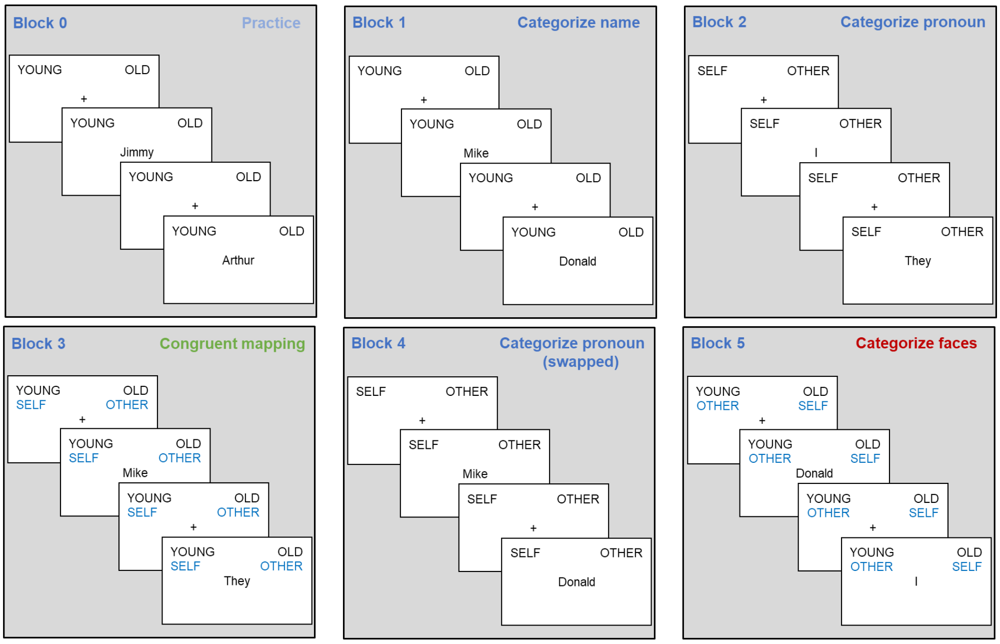
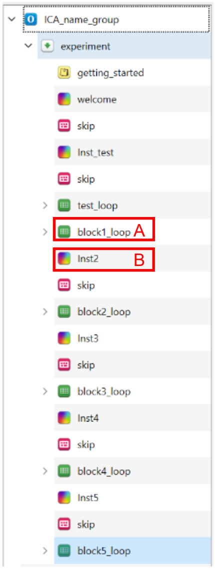


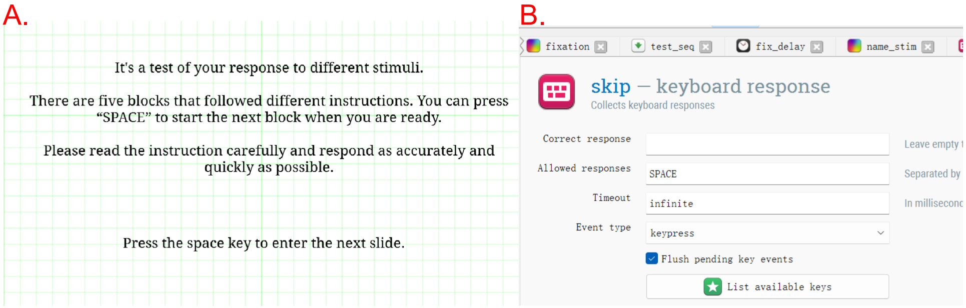
Figure 1. An overview of the four blocks of the implicit association task

Task introduction in OpenSesame

Task overview



ICA_name_group
experiment
getting_started
welcome
skip
Inst_test
skip
test_loop
block1_loop A
Inst2 B
skip
block2_loop
Inst3
skip
block3_loop
Inst4
skip
block4_loop
Inst5
skip
block5_loop



A.

It's a test of your response to different stimuli.

There are five blocks that followed different instructions. You can press "SPACE" to start the next block when you are ready.

Please read the instruction carefully and respond as accurately and quickly as possible.

Press the space key to enter the next slide.

B.

skip – keyboard response
Collects keyboard responses

Correct response:
Allowed responses: SPACE
Timeout: infinite
Event type: keypress
☑ Flush pending key events

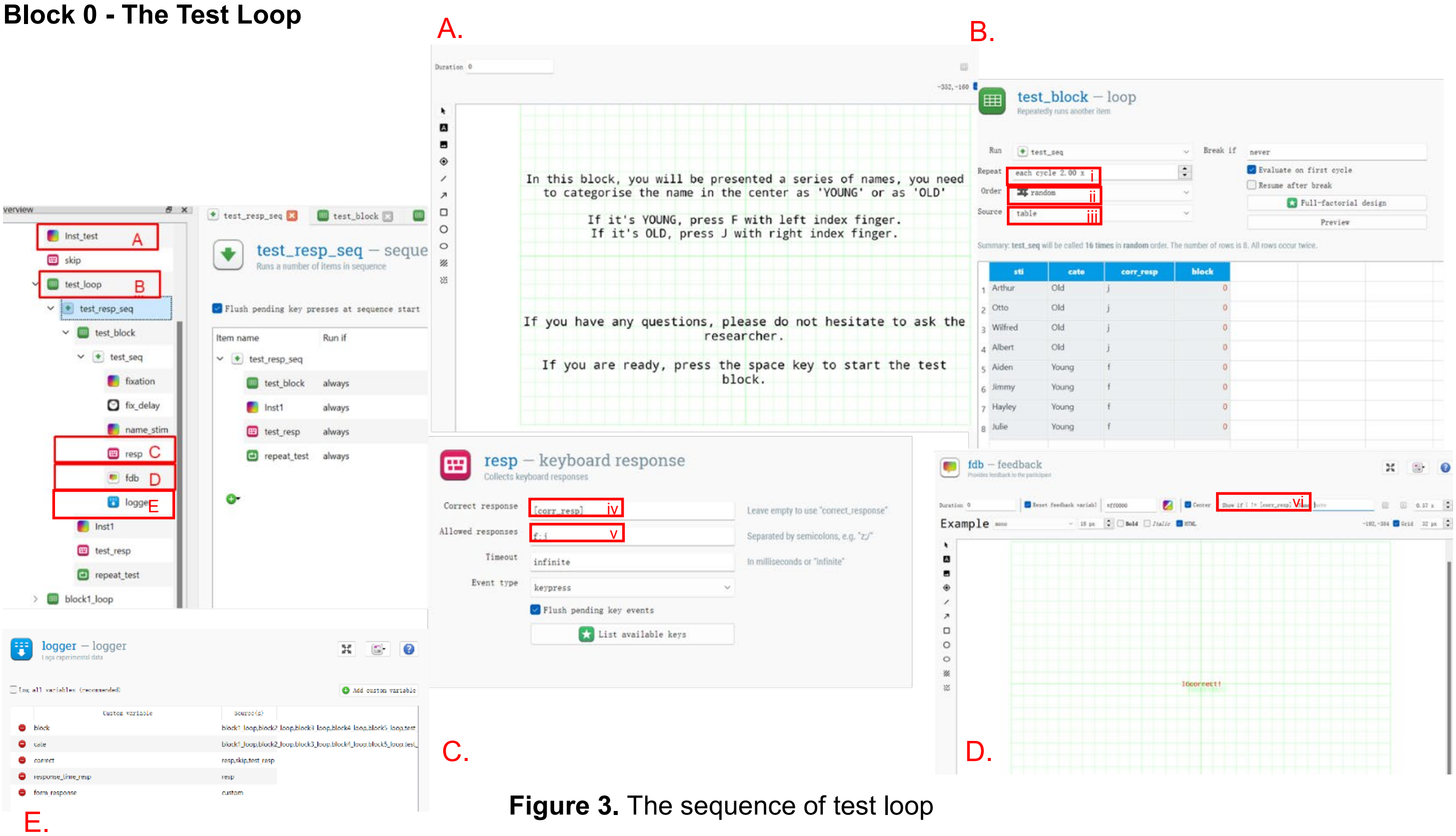
Figure 2. The overview area of OpenSesame and ***A.*** the welcome page and ***B.*** the skip setting.

This shows the profile of our design, I have six blocks in this experiment, including one practice group ("test_loop" in the picture) and five formal test blocks. At the beginning of the task, there is a "welcome" page (A) to show the participant a brief description of this task, and to give them more sense of control and freedom, the participants are told they can press the "space" key to skip this screen (B). Before each block, we have an instruction page to guide what they should do and emphasize what they should pay attention to. The same with the first block, they can skip the instruction and then enter the next block when they are ready by pressing the "space" key.

I use "test" and ordered numbers to name the loop and instruction page, which makes designers and the researcher in the future distinguish between blocks clearly and understand the flow of the task easier so that they could modify this conveniently on their purpose.

After we have an intuitive feeling of this task, let's start to go through the first loop – the test block.

Block 0 - The Test Loop



Duration 0

In this block, you will be presented a series of names, you need to categorise the name in the center as 'YOUNG' or as 'OLD'

If it's YOUNG, press F with left index finger.
If it's OLD, press J with right index finger.

If you have any questions, please do not hesitate to ask the researcher.

If you are ready, press the space key to start the test block.

☑ Flush pending key presses at sequence start

Item name	Run if
test_resp_seq	
test_block	always
Inst1	always
test_resp	always
repeat_test	always

test_block — loop
Repeatedly runs another item

Run: test_seq
Repeat: each cycle 2.00 x
Order: random
Source: table
Break if: never
☑ Evaluate on first cycle
□ Resume after break
Full-factorial design
Preview

Summary: test_seq will be called 16 times in random order. The number of rows is 8. All rows occur twice.

	sti	cate	corr_resp	block
1	Arthur	Old	j	0
2	Otto	Old	j	0
3	Wilfred	Old	j	0
4	Albert	Old	j	0
5	Aiden	Young	f	0
6	Jimmy	Young	f	0
7	Hayley	Young	f	0
8	Julie	Young	f	0

resp — keyboard response
Collects keyboard responses

Correct response: [corr_resp] — Leave empty to use "correct_response"
Allowed responses: f;j — Separated by semicolons, e.g. "z;/"
Timeout: infinite — In milliseconds or "infinite"
Event type: keypress
☑ Flush pending key events
List available keys

fdb — feedback
Provides feedback to the participant

Duration 0 ☑ Reset feedback variabl #ff0000 ☑ Center Show if [!= [corr_resp]
Example mono 18 px □ Bold □ Italic ☑ HTML -192, -384 ☑ Grid 32 px

Incorrect!

logger — logger
Logs experimental data

□ Log all variables (recommended)
Add custom variable

Custom variable	Source(s)
block	block1_loop,block2_loop,block3_loop,block4_loop,block5_loop,test_
cate	block1_loop,block2_loop,block3_loop,block4_loop,block5_loop,test_
correct	resp,skip,test_resp
response_time_resp	resp
form_response	custom

Figure 3. The sequence of test loop

A test block is used to help participants familiarize themselves with these keys and respond to make sure they have fully understood this task.

(*A*) The "instruction" item shows participant how to respond, and remind them to ask a question if they have any questions. The duration of all the "sketchpads" is set to "0" because, after each sketchpad, I use the "response" component to manipulate the presenting time and define the response type.

(*B*) The "loop" item here stores all the stimulus and related information of them. (*i*) I choose to repeat twice by ticking "2.00x" in the "repeat" box, trying to make the practice more likely to be the same as the formal test but also do not take too much time. (*ii*) For the "order" blank, we choose "random", because we don't want the participant to remember the order, which might have a negative influence on their reaction time, as well as avoid the influence of specific order (e.g., the frequency and similarity of two names). In this part, (*iii*) we can see a table (because we choose the "source" as "table", you can also invoke from an external file), every column of the table means this trial will present once, with all the related information (e.g., the name of stimuli, category, correct response, and block) can be recorded and invoked in the same trial.

(*C*) This is a keyboard response component. In this section, we can (*iv*) "set the correct response" (here, we also call it from the loop table, putting the variable name within square brackets), and (*v*) set the allowed keys in the last page (to reduce the mistype by accident).

(*D*) In the feedback page, I input two "draw text line" elements, one is "correct", another is "incorrect". If we want this feedback correctly, we need to specify its parameter of the (*vi*) "show if" box. For the "Correct!" feedback, we set "show if [response_resp] == [corr_resp]", which means when participant's response to the stimuli is the same with the the matched value (in the same column) of the "corr_resp" variable (which we could see in the table of "loop"). I also set the color of this feedback to "green" and "read" respectively to make it more significant for participant.

(*E*) Finally, I attach a "logger" in the end of the block, to log all the collected response and related information. In the logger part, I set custom variable, to make my results clearer and easy to follow.

IAT task: Block 1 ~ Block 5

For Block 1 to Block 5, they are all similar with Block 0 (the test loop) with (1) fixation (sketchpad); (2) fix_delay (advanced delay); (3) xxx_stim (sketchpad); (4) response (keyboard response); (5) fdb (feedback); (6) logger (logger). For these blocks, the components 1,2,4,5 are linked, sharing the same parameter. However, they are under different loops, so the values they invoke are different that they can achieve different functions and aims.

For the stimulation sketchpad (component 3, Figure 4), because the lead words are in the left/right upper, and it

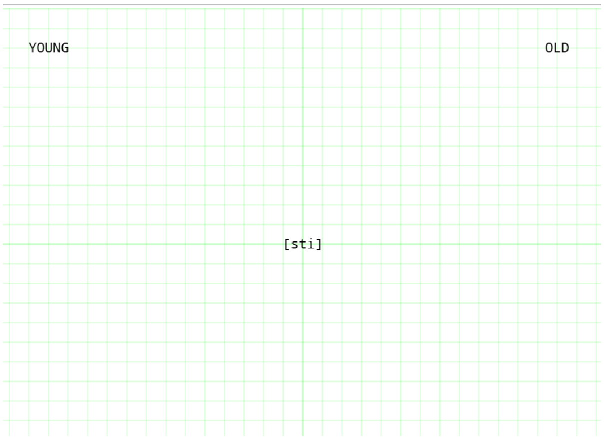
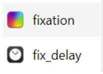


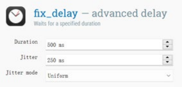
Figure 4. The stimulation sketchpad in OpenSesame

changes only across the block, I make a new sketchpad and put them in the right places. I use "[]" (square brackets) again to load the variables in the loop table just like I mentioned above.

In the mix block (block 3/5), I also set the stimulation's "color" to [color], to present the corresponding color as the instruction.

Besides, with the combination of fixation and fix_delay, (see in Figure 2), the duration of fixation is set to be "radom" , with the mean of 500 ms and the jitter of 250 ms in uniform distribution.

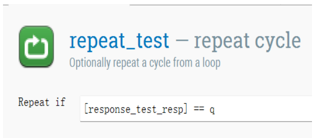




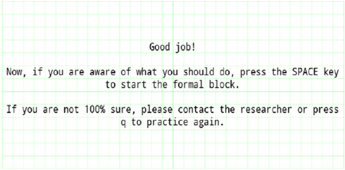
Duration 500 ms
Jitter 250 ms
Jitter mode Uniform

Highlights (Innovation)

Above all are the basic part of the IAT, besides, I also use some practical functions to make my task more completed. For example, This is widely used in psychological tasks that measure reaction time, because if we use a fixed duration of fixation, there would be a habitual response and expectation effect, which might negatively affect the behavioural results and it's likely to fail to get the effect we want (it could also be quite useful to reduce a simple-visual-stimulation-induced potential if you are going to record EEG data simultaneously). Also, I put a repeat cycle in the end of the test (also see in Figure 2) and a instruction in test loop, which allows participants to press "q" to practice again if they want to familiarize the keyboards and the task again before the formal test. Last but not least, remember, If we want to use this task in a real experiment, we need to make a second version, (which the order of blocks should chang – in accordance to the order of block 4 and 5, then 2 and 3).


repeat_test — repeat cycle
Optionally repeat a cycle from a loop

Repeat if [response_test_resp] == q

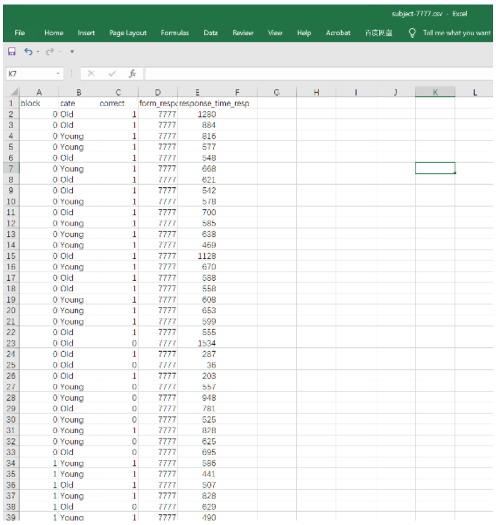


Good job!

Now, if you are aware of what you should do, press the SPACE key to start the formal block.

If you are not 100% sure, please contact the researcher or press q to practice again.

Results



	A	B	C	D	E
1	block	cate	correct	form_resp	response_time_resp
2	0	Old	1	7777	1280
3	0	Old	1	7777	884
4	0	Young	1	7777	816
5	0	Young	1	7777	577
6	0	Old	1	7777	548
7	0	Young	1	7777	668
8	0	Old	1	7777	621
9	0	Old	1	7777	542
10	0	Young	1	7777	578
11	0	Old	1	7777	700
12	0	Young	1	7777	585
13	0	Young	1	7777	638
14	0	Young	1	7777	469
15	0	Old	1	7777	1128
16	0	Young	1	7777	670
17	0	Old	1	7777	588
18	0	Old	1	7777	558
19	0	Young	1	7777	608
20	0	Young	1	7777	653
21	0	Young	1	7777	599
22	0	Old	1	7777	555
23	0	Old	0	7777	1534
24	0	Old	1	7777	287
25	0	Old	0	7777	36
26	0	Old	1	7777	203
27	0	Young	0	7777	557
28	0	Young	0	7777	948
29	0	Old	0	7777	781
30	0	Young	0	7777	525
31	0	Young	1	7777	828
32	0	Young	0	7777	625
33	0	Old	0	7777	695
34	1	Young	1	7777	586
35	1	Young	1	7777	441
36	1	Old	1	7777	507
37	1	Young	1	7777	828
38	1	Old	0	7777	629
39	1	Young	1	7777	490

Figure 5 The results of subject 1 in ".csv" file.

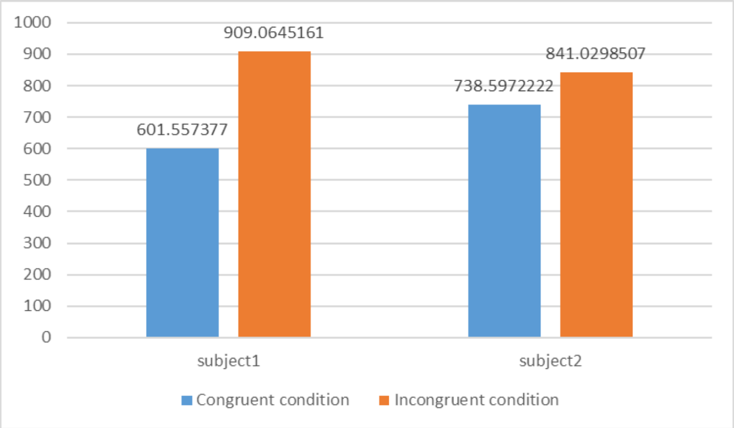
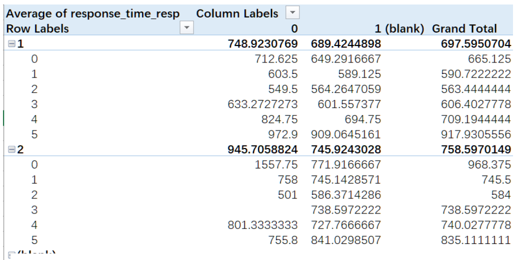


Figure 7 It's easier to categorize words and faces if the categories POSITIVE/YOUNG and NEGATIVE/OLD are combined (as compared to the reverse).



Average of response_time_resp Row Labels	Column Labels 0	1	(blank)	Grand Total
⊟1	**748.9230769**	**689.4244898**		**697.5950704**
0	712.625	649.2916667		665.125
1	603.5	589.125		590.7222222
2	549.5	564.2647059		563.4444444
3	633.2727273	601.557377		606.4027778
4	824.75	694.75		709.1944444
5	972.9	909.0645161		917.9305556
⊟2	**945.7058824**	**745.9243028**		**758.5970149**
0	1557.75	771.9166667		968.375
1	758	745.1428571		745.5
2	501	586.3714286		584
3		738.5972222		738.5972222
4	801.3333333	727.7666667		740.0277778
5	755.8	841.0298507		835.1111111

Figure 6 The pivot table of 2 subjects.

I collect two data from two different people, then put them together and use the pivot table to calculate. Then I subtract congruent condition (block 3) to incongruent condition (block 5) for the correct response trials.

For subject 1, the ageist bias is 601.56 - 909.06 = - 307.50.

For subject 2, the ageist bias is 738.60 - 841.03 = - 102.43.

In this case, both subjects have ageist bias, and the subject 1 might have more bias than subject 2.