Contextual Task (TL) SHAPING PLAN

*(cortex.cfg must be set to 8 bits per pixel, and eog\_mapping parameters must be set to 0 - parameters gain should be set to 2)*

**matlab scripts:**

colorarrayTT1.m - makes single T sets and item files, no L, T is full-bright

colorarrayTN.m - makes sets with T full-bright and dim L’s in any number

brightness of L’s can be increased relative to 0,1

Tlreactiontime.m - analysis per block of reaction time for repeated (trial type 1) vs new (trial type 2)

**Calibration task:** goes to extreme points of the receive screen, X+- 10DVA, Y+-8DVA

C:\cortex\contx\clrcalTL1.tim , clrcalTL.cnd, clrcalTL.itm

**Contextual task:** final version timing is 700ms T-fixation , required fixation for fixation spot is 500ms, 15 sec max scanning time, fixwin size around T is 6\*, T size is 1.74\*, min T-fixation window is ~4\*, around center + is 3\*, and num rewards can vary per animal

**c:\cortex\contx\tl.sav tl1.tim, tl.cnd, tl.blk**

**Training steps:** tl1.tim T-fixation timing can be changed as needed

TT009.itm T only

TN004.itm T + 3 dim L’s (0.1, 0.2) – do not use this step

TN005.itm T + 4 dim L’s (0.1, 0.2) – do not use this step

TN006.itm T + 5 dim L’s (0.1, 0.2) – do not use this step

TN007.itm T + 6 dim L’s (0.1, 0.2) – do not use this step

TN008.itm T + 7 dim L’s (0.1, 0.2) – do not use this step

TN010.itm T + 8 dim L’s (0.1, 0.2) – do not use this step

TN011.itm T + 12 dim L’s (0.1, 0.2) – do not use this step

TN012.itm T + 3 dim L’s (0.1, 0.3)

TN013.itm T + 4 dim L’s (0.1, 0.4)

TN014.itm T + 4 dim L’s (0, 0.4)

TN015.itm T + 5 dim L’s (0, 0.5)

TN016.itm T + 6 dim L’s (0, 0.6)

TN017.itm T + 7 dim L’s (0, 0.7)

TN018.itm T + 8 dim L’s (0, 0.8)

TN019.itm T + 9 dim L’s (0, 0.9)

TN020.itm T + 10 L’s (0, 1) - L’s are at full bright

\*\*\*\*All of the following are isoluminant and use the random number generator\*\*\*\*

**OLD Tloc and OLD TL.itm files used prior to 6/18/10**

**DO NOT ANALYSE DATA FROM SETS: TL400-417 (run between 6/19/10-5/24/11) – TLOC FOLDER OVERWRITTEN. NEW TL400-417 made beginning 5/24/11**

***TL021-TL030 can be skipped***

**TL021-TL030.itm** 20 block (12 repeated, 12 new per block)final task NO offset (generated with **colorarray1.m**) – *use with tt.sav (tl1.tim, tl.cnd, tl.blk)*

**TL2.sav (tl1.tim, tl2.cnd, tl2.blk)**

**TL031– TL050.itm** 60 block(12 repeated, 12 new per block)NO offset(generate with **colorarray2.m**) - *~~use with tl2.sav (tl1.tim, tl2.cnd, tl2.blk)~~*

**TL051– TL099.itm** 60 block(12 repeated, 12 new per block)EASIEST offset **(.55, .05)** (generate with **TL offset 1.m**) - *Not a necessary training step for most animals*

**TL100– TL199.itm** 60 block(12 repeated, 12 new per block)offset **(.5, .1)** (generate with **TL offset 2.m**) - *use with tl2.sav (tl1.tim, tl2.cnd, tl2.blk)*

**TL200– TL299.itm** 60 block(12 repeated, 12 new per block)offset **(.45, .15)** (generate with **TL offset 3.m**) - *use with tl2.sav (tl1.tim, tl2.cnd, tl2.blk)*

**TL300– TL399.itm** 60 block(12 repeated, 12 new per block)offset **(.4, .2)** (generate with **TL offset 4.m**) - *use with tl2.sav (tl1.tim, tl2.cnd, tl2.blk)*

**TL400– TL499.itm** 60 block(12 repeated, 12 new per block)HARDEST offset **(.35, .25)** (generate with **TL offset 5.m**) - *use with tl2.sav (tl1.tim, tl2.cnd, tl2.blk)*

*VARIATIONS:*

**TLm - longer blocks**

**TLm.sav ( tl1.tim, tlm.cnd, tlm.blk)**

**TLm100– TLm199.itm** 30 block(24 repeated, 24 new per block)offset **(.5, .1)** (generate with **TLm offset 2.m**) - *~~use with tl3.sav (tl1.tim, tl3.cnd,??.blk)~~*

**TLm200– TLm299.itm** 30 block(24 repeated, 24new per block)offset **(.45, .15)** (generate with **TLm offset 3.m**) - *~~use with tl3.sav (tl1.tim, tl3.cnd, ??.blk)~~*

**TLm300– TLm399.itm** 30 block(24 repeated, 24 new per block)offset **(.4, .2)** (generate with **TLm offset 4.m**) - *~~use with tl3.sav (tl1.tim, tl3.cnd, ??.blk)~~*

**TLm400– TLm499.itm** 30 block(24 repeated, 24 new per block)HARDEST offset **(.35, .25)** (generate with **TLm offset 5.m**) - *~~use with tl3.sav (tl1.tim, tl3.cnd, tl2.blk)~~*

**TLc - at block 20, repeated context T location switched with L position of same color**

**TLc.sav (tl1.tim, tlc.cnd, tlc.blk)**

**TLc100– TLc199.itm** 30 block(24 repeated, 24 new per block)offset **(.5, .1)** (generate with **TLc offset 2.m**) - *~~use with tl3.sav (tl1.tim, tl3.cnd,??.blk)~~*

**TLc200– TLc299.itm** 30 block(24 repeated, 24new per block)offset **(.45, .15)** (generate with **TLc offset 3.m**) - *use with tl3.sav (tl1.tim, tl3.cnd, ??.blk)*

**TLc300– TLc399.itm** 30 block(24 repeated, 24 new per block)offset **(.4, .2)** (generate with **TLc offset 4.m**) - *use with tl3.sav (tl1.tim, tl3.cnd, ??.blk)*

**TLc400– TLc499.itm** 30 block(24 repeated, 24 new per block)HARDEST offset **(.35, .25)** (generate with **TLc offset 5.m**) - *use with tl3.sav (tl1.tim, tl3.cnd, tl2.blk)*

**TLx - “new” for each block has L’s in same position as previous blocks, but new T location**

**TLx.sav (tl1.tim, tlx.cnd, tlx.blk)**

**TLx100– TLx199.itm** 36 block(12 repeated, 12 new per block)offset **(.5, .1)** (generate with **TLx.m**) - *~~use with tlx.sav (tl1.tim, tl3.cnd,??.blk)~~*

**TLx200– TLx299.itm** 36 block(12 repeated, 12 new per block)offset **(.45, .15)** (generate with **TLx.m**) - *use with tlx.sav (tl1.tim, tlx.cnd, ??.blk)*

**TLx300– TLx399.itm** 36 block(12 repeated, 12 new per block)offset **(.4, .2)** (generate with **TLx.m**) - *use with tlx.sav (tl1.tim, tlx.cnd, ??.blk)*

**TLx400– TLx499.itm** 36 block(12 repeated, 12 new per block)HARDEST offset **(.35, .25)** (generate with **TLx.m**) - *use with tlx.sav (tl1.tim, tlx.cnd, tl2.blk)*

**TLs - “predictable” are repeated exactly, “unpredictable” change only the Tloc, “new” are entirely novel**

**TLx100– TLx199.itm** 36 blocks(12 predictable, 1 unpredictable, 12 new per block)offset **(.5, .1)** (generate with **TLx.m**) - *~~use with tls.sav (tl1.tim, tl3.cnd,??.blk)~~*

**TLx200– TLx299.itm** 36 blocks(12 predictable, 1 unpredictable, 12 new per block)offset **(.45, .15)** (generate with **TLx.m**) - *use with tls.sav (tl1.tim, tls.cnd, ??.blk)*

**TLx300– TLx399.itm** 36 blocks(12 predictable, 1 unpredictable, 12 new per block)offset **(.4, .2)** (generate with **TLx.m**) - *use with tls.sav (tl1.tim, tls.cnd, ??.blk)*

**TLx400– TLx499.itm** 36 blocks(12 predictable, 1 unpredictable, 12 new per block) HARDEST offset **(.35, .25) .** (generate with **TLx.m**) - *use with tls.sav (tl1.tim, tls.cnd, tl2.blk)*

**TLbw - itmes in all contexts are gray beginning at block ## (i.e. 13 or 21)**

**##bw100– ##bw199.itm:** 60 blocks(12 old, 12 new per block)offset **(.5, .1)** (generate with **TLx.m**) - *~~use with tls.sav (tl1.tim, tl3.cnd,??.blk)~~*

**##bw200– ##bw 299.itm:** 60 blocks(12 old, 12 new per block)offset **(.45, .15)** (generate with **TLx.m**) - *use with tls.sav (tl1.tim, tls.cnd, ??.blk)*

**##bw 300– ##bw 399.itm:** 60 blocks(12 old, 12 new per block)offset **(.4, .2)** (generate with **TLx.m**) - *use with tls.sav (tl1.tim, tls.cnd, ??.blk)*

**##bw 400– ##bw 499.itm:** 60 blocks(12 old, 12 new per block) HARDEST offset **(.35, .25) .** (generate with **TLx.m**) - *use with tls.sav (tl1.tim, tls.cnd, tl2.blk)*

**TLshape (O’s and ovals) TL.sav (tl1.tim, tl.cnd, tl.blk - 20 blks)**

**.itm NAME SCENE OVAL INTENSITY BLOCKS**

**TLsh001 O (.65, .65) 20**

**TLsh002 O + 3 (.3, .8) (.1, .3) 20**

**TLsh003 O + 4 (.3, .8) (.1, .4) 20**

**TLsh004 O + 5 (.3, .8) (.1, .5) 20**

**TLsh005 O + 6 (.3, .8) (0, .6) 20**

**TLsh006 O + 7 (.3, .8) (0, .7) 20**

**TLsh007 O + 8 (.3, .8) (0, .8) 20**

**TLsh008 O + 9 (.3, .8) (0, .9) 20**

**TLsh009 O + 10 (.3, .8) (0, 1) 20**

**TLsh010 O + 11 (.3, .8) (0, 1) 20**

**TLsh011 - 015 O + 11 (.3, .8) isoluminant 60 (tl2.sav)**

**TLsh100 O + 11 (.4, .75) isoluminant 60**

TLsh200 O + 11 (.5, .7) isoluminant 60

TLsh300 O + 11 (.6, .7) isoluminant 60

##sh( )00 O+11 (.6, .7) isoluminant 60

**\*\*shape changes from T&L to O&0 at block ##**

**() indicates T&L offset [O&0+100]**

**\*\*\***no bold**indicate sets not yet made**