https://zenodo.org/record/7324830

Supplementary material for the paper "The DRM illusion in short-term memory: Opposite effects of retention interval on true and false recognition"

Guillermo Campoy (gcampoy@um.es)

Abstract of the paper

A short-term memory (STM) version of the Deese/Roediger-McDermott (DRM) paradigm was employed to investigate how true and false recognition evolved as STM contents were lost over a short time-window immediately after initial encoding. Presentation of six-word DRM lists were followed by list-specific recognition tests applied either immediately or after a distractor-filled retention interval of 3, 9, or 27 s. Results showed that the probability of true recognition decreased, and the probability of false recognition increased as the retention interval grew longer. Based on the fuzzy-trace theory, we suggest that this pattern emerged from the different durability in STM of item-specific phonological representations, which would play the dual role of supporting true memory and preventing false recognition, and integrative semantic representations, whose overlap with the critical items would give rise to the DRM illusion.

Keywords: False memory, recognition, short-term memory, DRM, fuzzy-trace theory, explicit warnings, Bayesian multilevel modeling.

Columns of the file “ONE ROW FOR EACH TEST ITEM”

· participant_id (from 1 to 50)

· number_of_trial (from 1 to 40)

· number_list_items_at_test (from 1 to 5)

· interval_in_seconds (0, 3, 9, or 27 s)

· interval_in_1234 (1 = 0 s, 2 = 3 s, 3 = 9 s, 4 = 27 s)

· type_of_item (list word, critical item, or unrelated distractor)

· treatment (from 1 to 12, according to the following table)

	Retention interval			
Type of item	0 s	3 s	9 s	27 s
List word	1	2	3	4
Critical item	5	6	7	8
Unrelated distractor	9	10	11	12

· serial_position (from serial position 1 to serial position 6; 0 = non-presented, unrelated distractor; 7 = non-presented, critical item)

· presentation_frame (presentation frame at recognition test, numbered from top to bottom and left to right. From 1 to 9)

· item (the specific Spanish word)

· item_id (from 1 to 480)

· times_clicked (note that a word could be clicked twice to unselected it)

· finally_clicked (the final state of the word; 1 = selected; 0 = non-selected)

Columns of the file "MODEL SOURCE FILE" (subset of the previous file)

· participant_id (from 1 to 50)

· treatment (from 1 to 12)

· item_id (from 1 to 480)

· finally_clicked (the final state of the word; 1 = selected; 0 = non-selected)

https://zenodo.org/record/7324830

Columns of the file "ONE ROW FOR EACH TRIAL"

· participant_id (from 1 to 50)

· gender (0 = female, 1 = male)

· age

· number_of_trial (from 1 to 50)

· interval_in_seconds (0, 3, 9, or 27 s)

· number_list_items_at_test (from 1 to 5)

· number_unrelated_distractors_at_test (from 3 to 7)

· list_item_1 (list word presented in serial position 1)

· list_item_2 (list word presented in serial position 2)

· list_item_3 (list word presented in serial position3)

· list_item_4 (list word presented in serial position 4)

· list_item_5 (list word presented in serial position 5)

· list_item_6 (list word presented in serial position 6)

· critical_item

· test_item_1 (test item presented in frame 1)

· test_item_2 (test item presented in frame 2)

· test_item_3 (test item presented in frame 3)

· test_item_4 (test item presented in frame 4)

· test_item_5 (test item presented in frame 5)

· test_item_6 (test item presented in frame 6)

· test_item_7 (test item presented in frame 7)

· test_item_8 (test item presented in frame 8)

· test_item_9 (test item presented in frame 9)

· clicked_word_1 (first clicked word)

· clicked_word_2 (second clicked word)

· clicked_word_3 (third word clicked)

(and so on… Note that a word could be clicked twice to unselected it)

https://zenodo.org/record/7324830

R script to fit the Bayesian multilevel model

```r
# Read source data
d <- read.csv2("model_source_file.csv")

# INFO: Values of 'treatment' in d
#
#                           INTERVAL
# TYPE OF ITEM           0S   3S   9S   27S
# list word               1    2    3    4
# critical item           5    6    7    8
# unrelated distractor    9   10   11   12

# Load the required library 'rethinking'
# McElreath, R. (2020). Statistical rethinking: A Bayesian course with
examples in R and Stan, second edition. Boca Raton, FL: Chapman and Hall/CRC.

library (rethinking)

# Create the dat object
dat <-list(
  resp = as.integer(d$finally_clicked),
  treatment = as.integer(d$treatment),
  actor = as.integer(d$participant_id),
  item = as.integer(d$item_id)
)

# Fit the model via Stan
# See McElreath, 2020, model m14.3, p. 449
m <-ulam(
  alist(
    resp ~binomial(1,p),
    logit(p) <- g[treatment]+alpha[actor,treatment]+beta[item,treatment],

    # adaptive priors - non-centered
    transpars> matrix[actor,12]:alpha<-
      compose_noncentered(sigma_actor,L_Rho_actor,z_actor),
    transpars> matrix[item,12]:beta<-
      compose_noncentered(sigma_item,L_Rho_item,z_item),

    matrix[12,actor]:z_actor ~normal(0,1.5),
    matrix[12,item]:z_item ~normal(0,1.5),

    # fixed priors
    g[treatment] ~normal(0,1.5),
    vector[12]:sigma_actor ~dexp(1),
    cholesky_factor_corr[12]:L_Rho_actor ~lkj_corr_cholesky(2),
    vector[12]:sigma_item ~dexp(1),
    cholesky_factor_corr[12]:L_Rho_item ~lkj_corr_cholesky(2),


    # compute ordinary correlation matrixes from Cholesky factors
    gq> matrix[12,12]:Rho_actor<<-Chol_to_Corr(L_Rho_actor),
    gq> matrix[12,12]:Rho_item<<-Chol_to_Corr(L_Rho_item)

  ) ,data=dat,chains=4,cores=4,iter=6000, log_lik=FALSE)
```

R script to generate values showed in Table 1

```
samples <- extract.samples(m, n=12000)$g
samplesP <- as.data.frame(inv_logit(samples))

biasCorr1 <- samplesP[,c(1:4)] - samplesP[,c(9:12)]
biasCorr2 <- samplesP[,c(5:8)] - samplesP[,c(9:12)]

#Estimates of the bias corrected probability of true recognition
round(c(mean(biasCorr1$V1), mean(biasCorr1$V2), mean(biasCorr1$V3),
mean(biasCorr1$V4)), 2)

round (c(HPDI(biasCorr1$V1, prob=0.97), HPDI(biasCorr1$V2, prob=0.97),
HPDI(biasCorr1$V3, prob=0.97), HPDI(biasCorr1$V4, prob=0.97)), 2)

#Estimates of the bias corrected probability of false recognition
round( c(mean(biasCorr2$V5), mean(biasCorr2$V6), mean(biasCorr2$V7),
mean(biasCorr2$V8)), 2)

round (c(HPDI(biasCorr2$V5, prob=0.97), HPDI(biasCorr2$V6, prob=0.97),
HPDI(biasCorr2$V7, prob=0.97), HPDI(biasCorr2$V8, prob=0.97)), 2)
```

Table 1: Bayesian estimates of the bias-corrected probability of true and false recognition

	True recognition		False recognition	
Retention interval	M	97% HPDI	M	97% HPDI
0 s	0.90	[0.88, 0.93]	0.13	[0.07, 0.19]
3 s	0.90	[0.87, 0.93]	0.17	[0.10, 0.23]
9 s	0.86	[0.82, 0.89]	0.20	[0.13, 0.28]
27 s	0.82	[0.78, 0.86]	0.33	[0.24, 0.42]

Note: M, mean of the posterior distribution; HPDI, highest posterior density interval.

R script to generate non-corrected values (Table 1bis)

```
#Estimates of the probability of true recognition
round(c(mean(samplesP[,1]),mean(samplesP[,2]),mean(samplesP[,3]),mean(samplesP
[,4])), 4)

round(c(HPDI(samplesP[,1], prob=0.97),HPDI(samplesP[,2],
prob=0.97),HPDI(samplesP[,3], prob=0.97),HPDI(samplesP[,4], prob=0.97)),4)


#Estimates of the probability of recognition of critical items
round(c(mean(samplesP[,5]),mean(samplesP[,6]),mean(samplesP[,7]),mean(samplesP
[,8])), 4)

round(c(HPDI(samplesP[,5], prob=0.97),HPDI(samplesP[,6],
prob=0.97),HPDI(samplesP[,7], prob=0.97),HPDI(samplesP[,8], prob=0.97)),4)


#Estimates of the probability of recognition of unrelated distractors
round(c(mean(samplesP[,9]),mean(samplesP[,10]),mean(samplesP[,11]),mean(sample
sP[,12])), 4)

round(c(HPDI(samplesP[,9], prob=0.97),HPDI(samplesP[,10],
prob=0.97),HPDI(samplesP[,11], prob=0.97),HPDI(samplesP[,12], prob=0.97)),4)
```

Table 1bis: Bayesian estimates of the probability of recognition of list words, critical items, and unrelated distractors.

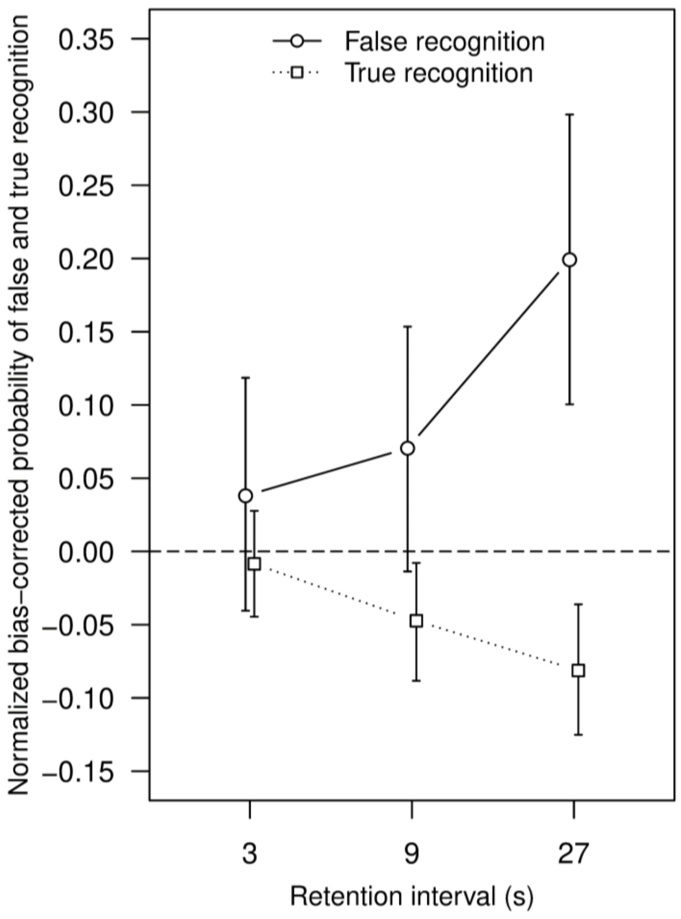
	List words		Critical items		Unrelated distractors	
Retention interval	M	97% HPDI	M	97% HPDI	M	97% HPDI
0 s	0.9054	[0.8822, 0.9279]	0.1307	[0.0745, 0.1922]	0.0013	[0.0002, 0.0029]
3 s	0.8972	[0.8662, 0.9261]	0.1690	[0.1047, 0.2365]	0.0016	[0.0003, 0.0034]
9 s	0.8608	[0.8266, 0.8988]	0.2039	[0.1344, 0.2797]	0.0041	[0.0011, 0.0076]
27 s	0.8262	[0.7837, 0.8677]	0.3318	[0.2436, 0.4320]	0.0033	[0.0009, 0.0066]

Note: M, mean of the posterior distribution; HPDI, highest posterior density interval.

R script to generate values depicted in Figure 2

```
normBiasCorr1 <- biasCorr1[,2:4] - biasCorr1[,1]
normBiasCorr2 <- biasCorr2[,2:4] - biasCorr2[,1]


round( c( mean(normBiasCorr1[,1]), mean(normBiasCorr1[,2]),
mean(normBiasCorr1[,3]) ),3)

round( c(HPDI(normBiasCorr1[,1], prob=0.97),HPDI(normBiasCorr1[,2],
prob=0.97),HPDI(normBiasCorr1[,3], prob=0.97)),3)

round( c( mean(normBiasCorr2[,1]), mean(normBiasCorr2[,2]),
mean(normBiasCorr2[,3]) ),3)

round( c(HPDI(normBiasCorr2[,1], prob=0.97),HPDI(normBiasCorr2[,2],
prob=0.97),HPDI(normBiasCorr2[,3], prob=0.97)),3)
```

Figure 2: Model estimates (posterior means and 97% HPDIs) of the bias-corrected probability of true and false recognition normalized with respect to the first retention interval (0 s). The x-axis is in log scale.


False recognition
True recognition
Normalized bias–corrected probability of false and true recognition
Retention interval (s)