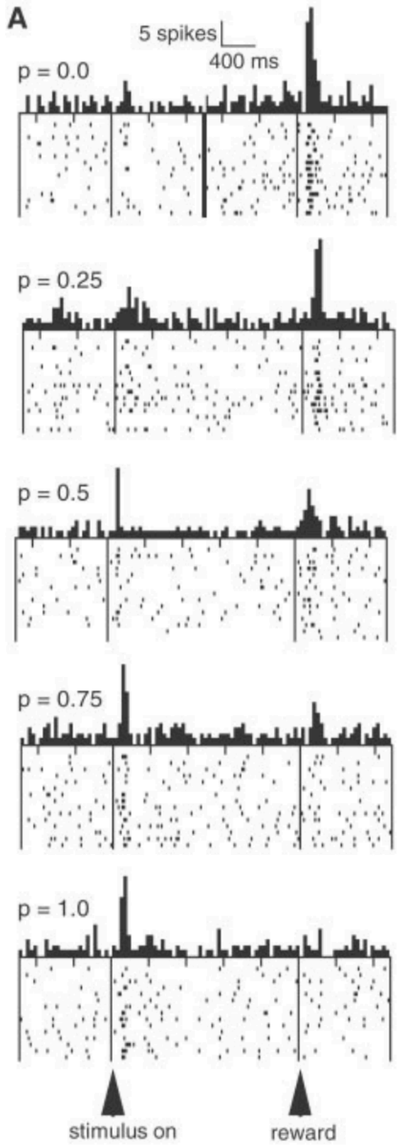
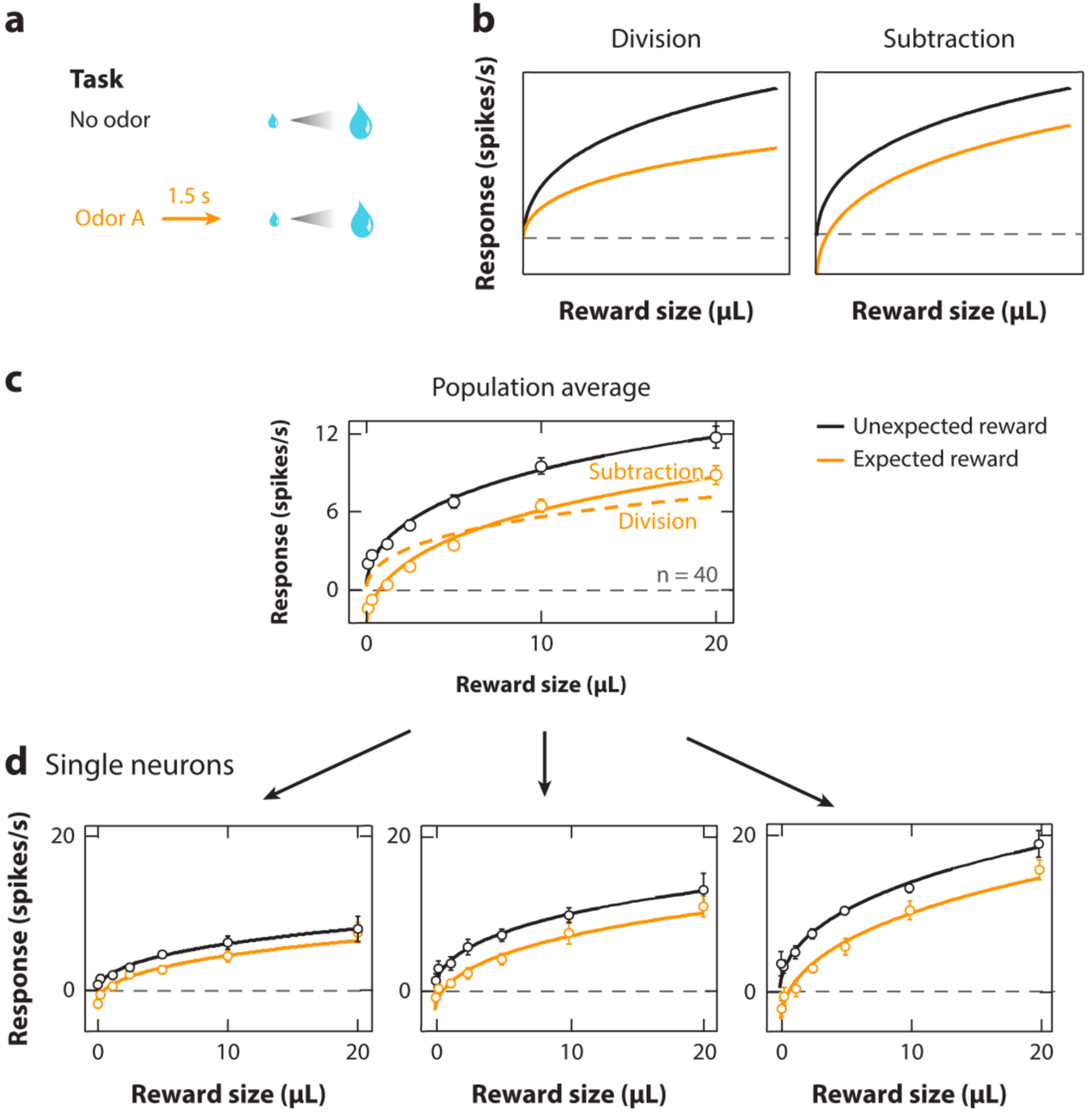
Distributional RL

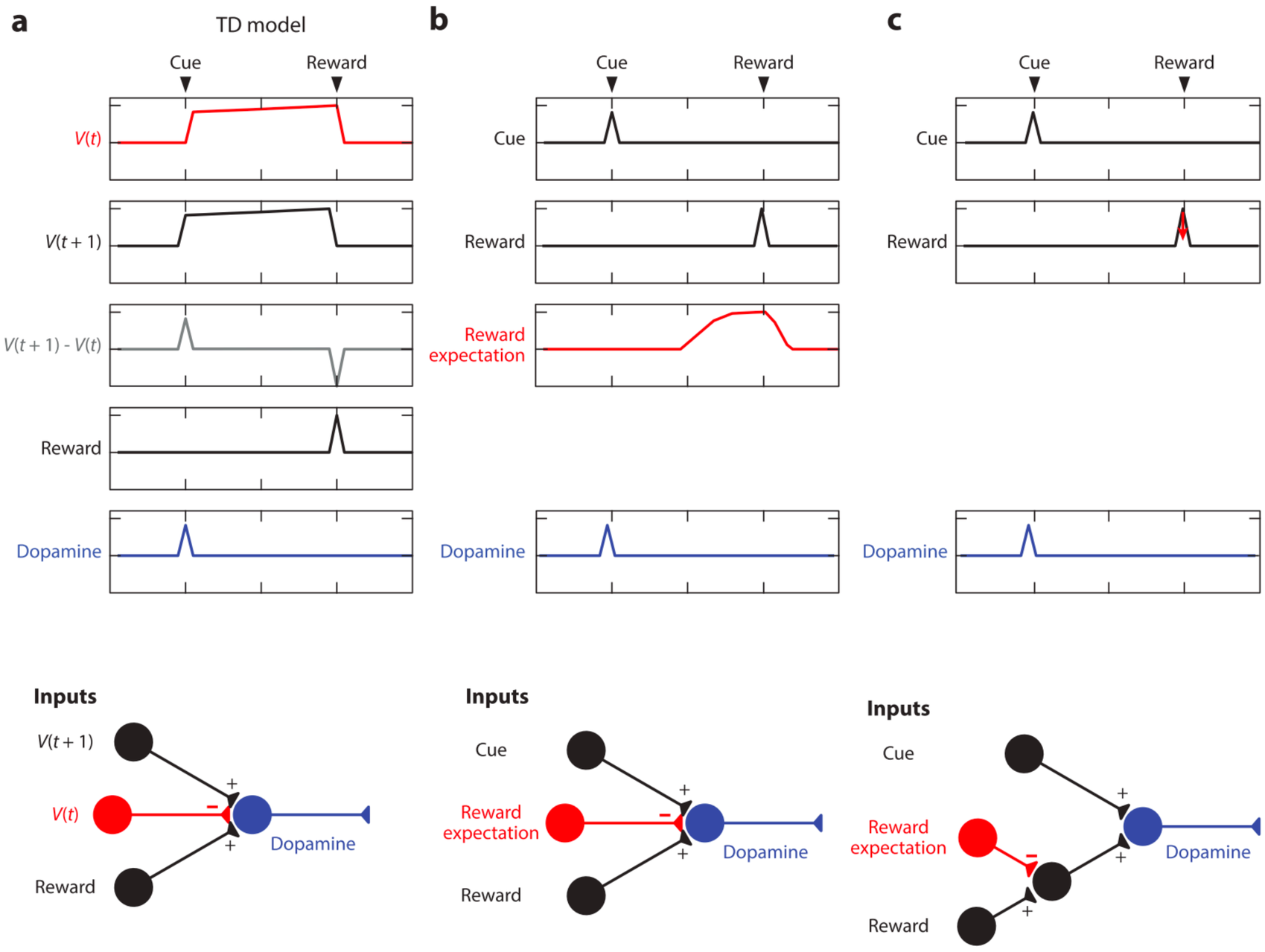
Brabeeba Wang
4/9/2021

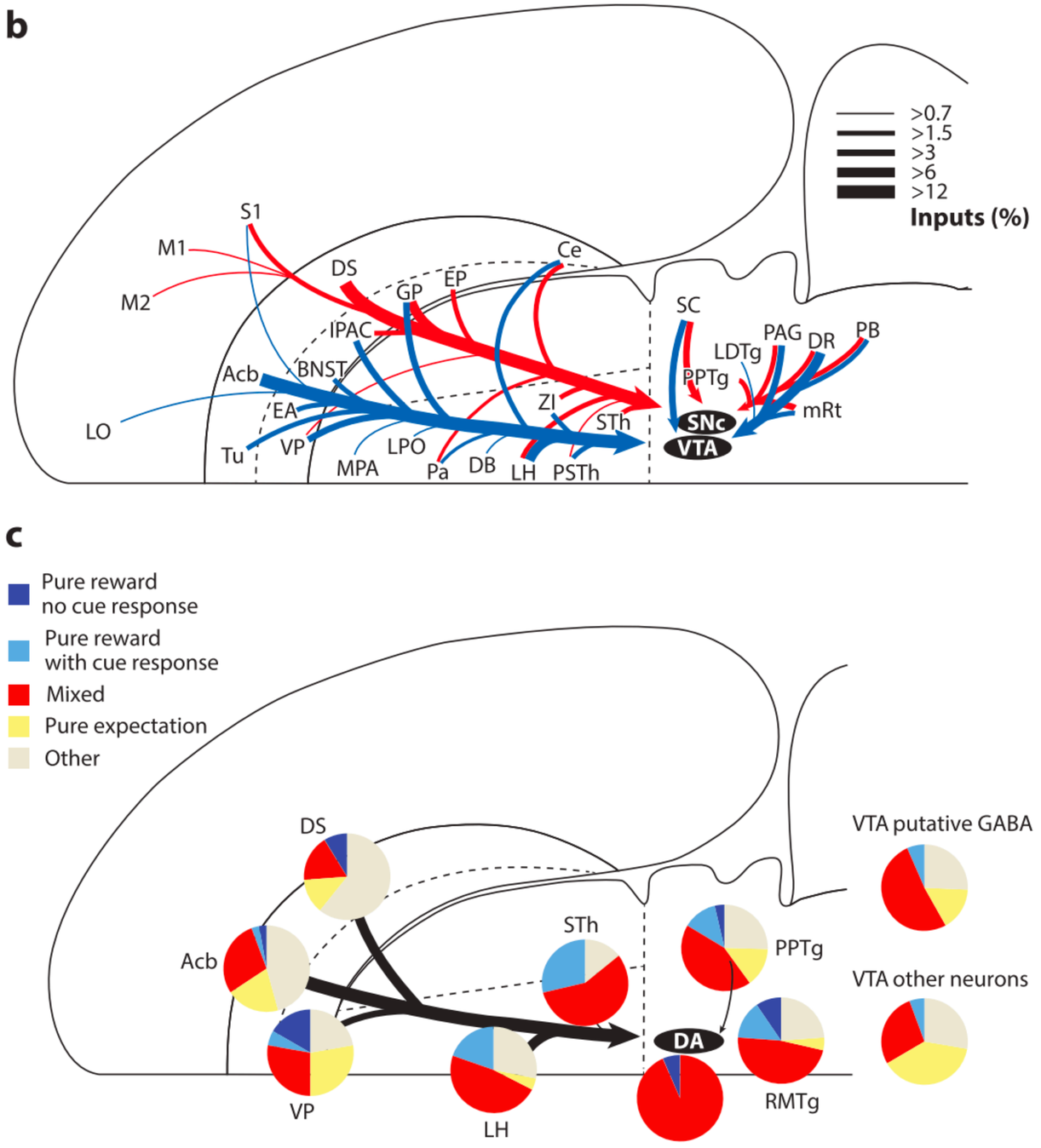
Dopamine

- One of the primary role of dopamine is to encode reward prediction error.
- $\delta = R(s) + \gamma V(s') - V(s)$
- This serves as a reinforcer for learning.
- By optogenetically activate DA neurons, one can reinforce desired behaviors in the animals




a
Task
No odor
Odor A
1.5 s
b
Division
Subtraction
Response (spikes/s)
Reward size (µL)
Reward size (µL)
c
Population average
Response (spikes/s)
12
6
0
Subtraction
Division
n = 40
0
10
20
Reward size (µL)
Unexpected reward
Expected reward
d
Single neurons
Response (spikes/s)
20
0
0
10
20
Reward size (µL)
Reward size (µL)
Reward size (µL)


a
TD model
Cue
Reward
V(t)
V(t + 1)
V(t + 1) - V(t)
Reward
Dopamine
b
Cue
Reward
Cue
Reward
Reward expectation
Dopamine
c
Cue
Reward
Cue
Reward
Dopamine
Inputs
V(t + 1)
V(t)
Reward
Dopamine
Inputs
Cue
Reward expectation
Reward
Dopamine
Inputs
Cue
Reward expectation
Reward
Dopamine


b
>0.7
>1.5
>3
>6
>12
Inputs (%)
S1
M1
M2
LO
Tu
Acb
EA
BNST
IPAC
DS
GP
EP
Ce
VP
MPA
LPO
Pa
DB
LH
PSTh
ZI
STh
SC
PPTg
LDTg
PAG
DR
PB
mRt
SNc
VTA
c
Pure reward
no cue response
Pure reward
with cue response
Mixed
Pure expectation
Other
DS
Acb
VP
STh
LH
PPTg
RMTg
DA
VTA putative GABA
VTA other neurons

(C)

Traditional RL

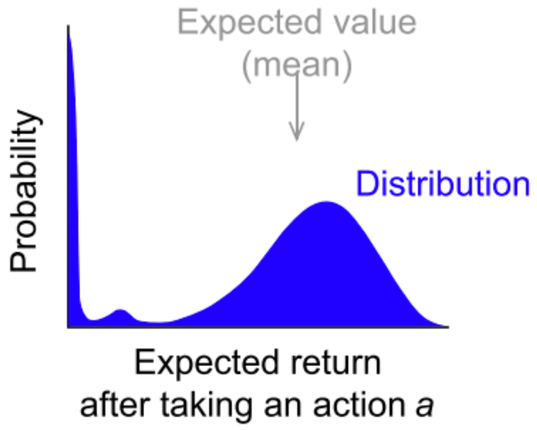
State

Expected value (mean)

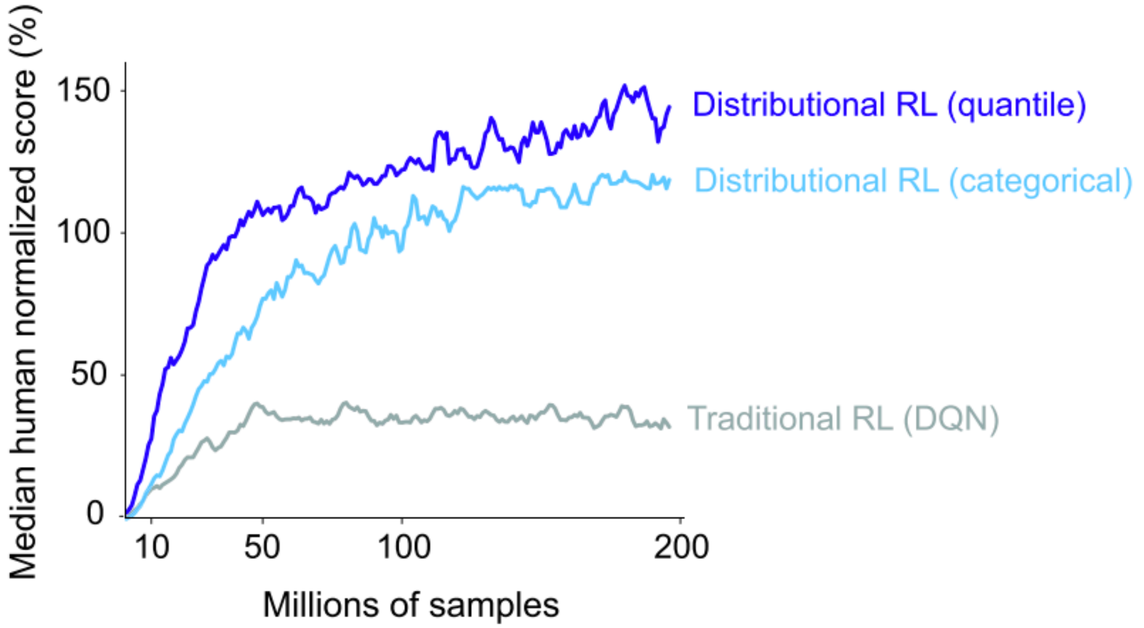
Distributional RL

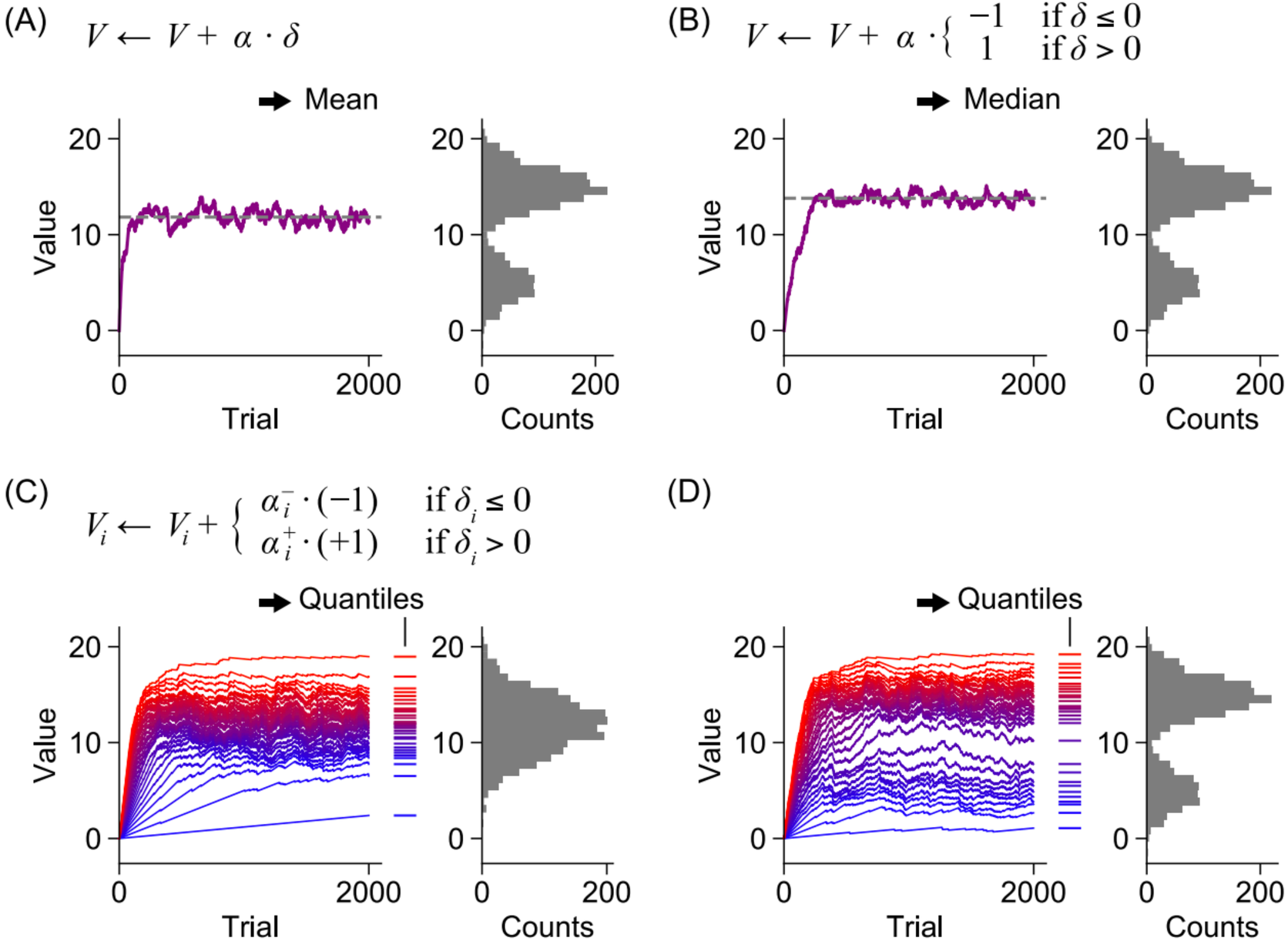
State

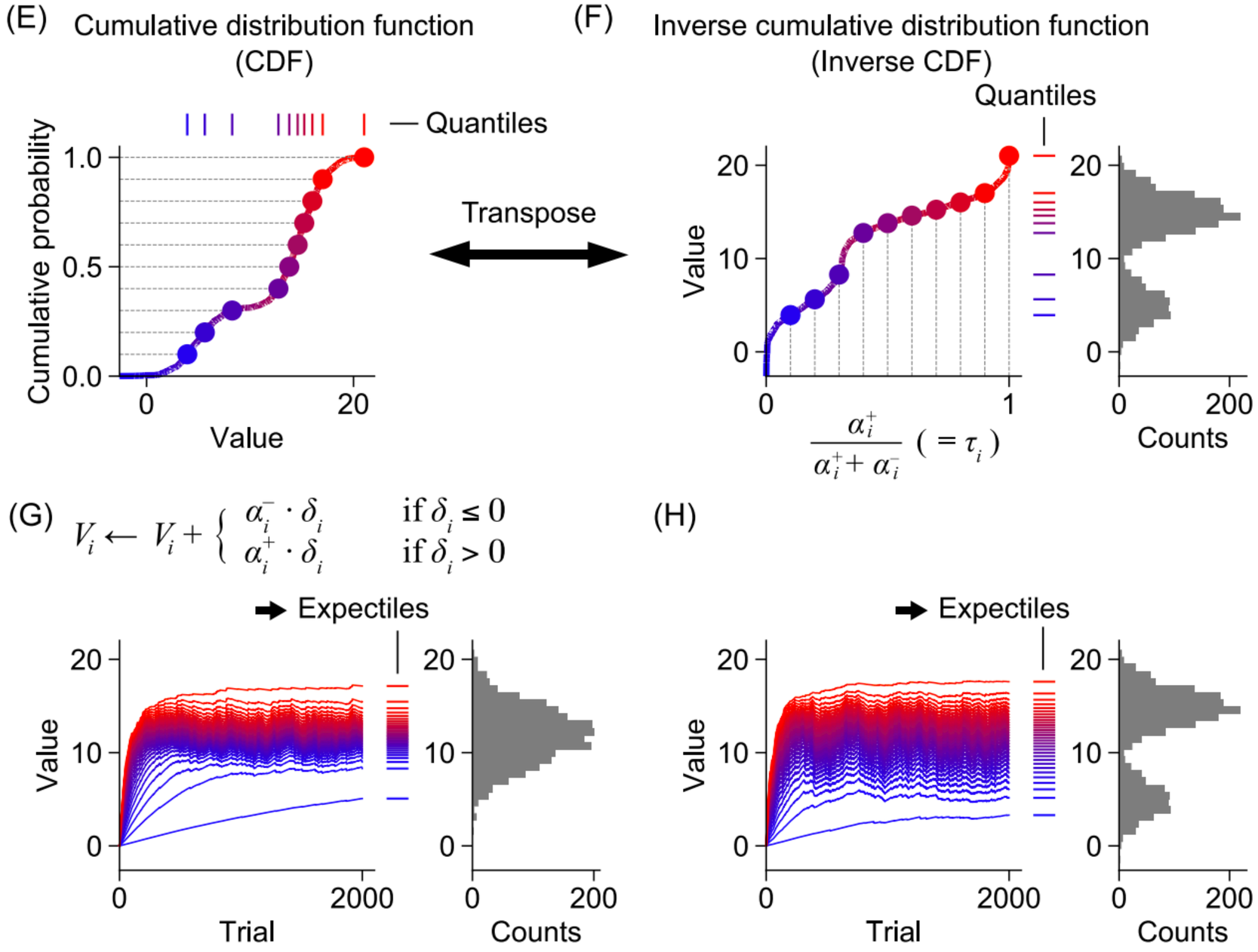
Distribution



(D)




(A)
$V \leftarrow V + \alpha \cdot \delta$
Mean
Value
Trial
Counts
(B)
$V \leftarrow V + \alpha \cdot \begin{cases} -1 & \text{if } \delta \leq 0 \\ 1 & \text{if } \delta > 0 \end{cases}$
Median
Value
Trial
Counts
(C)
$V_i \leftarrow V_i + \begin{cases} \alpha_i^- \cdot (-1) & \text{if } \delta_i \leq 0 \\ \alpha_i^+ \cdot (+1) & \text{if } \delta_i > 0 \end{cases}$
Quantiles
Value
Trial
Counts
(D)
Quantiles
Value
Trial
Counts

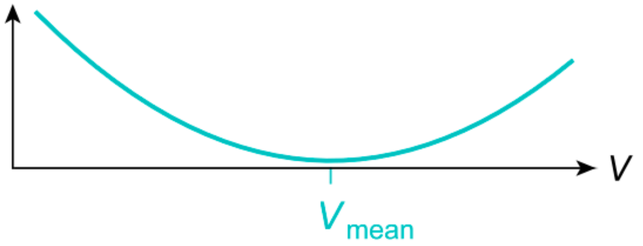

(E)
Cumulative distribution function
(CDF)
Quantiles
Cumulative probability
1.0
0.5
0.0
0
20
Value
Transpose
(F)
Inverse cumulative distribution function
(Inverse CDF)
Quantiles
Value
20
10
0
0
1
$\frac{\alpha_i^+}{\alpha_i^+ + \alpha_i^-}$ $(= \tau_i)$
20
10
0
0
200
Counts
(G)
$V_i \leftarrow V_i + \begin{cases} \alpha_i^- \cdot \delta_i & \text{if } \delta_i \le 0 \\ \alpha_i^+ \cdot \delta_i & \text{if } \delta_i > 0 \end{cases}$
Expectiles
Value
20
10
0
0
2000
Trial
20
10
0
0
200
Counts
(H)
Expectiles
Value
20
10
0
0
2000
Trial
20
10
0
0
200
Counts

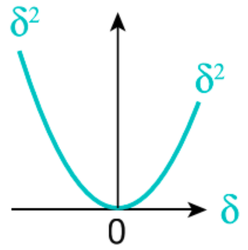
Loss (error) function

$$\delta_n = r_n - V$$

(B) Mean squared error

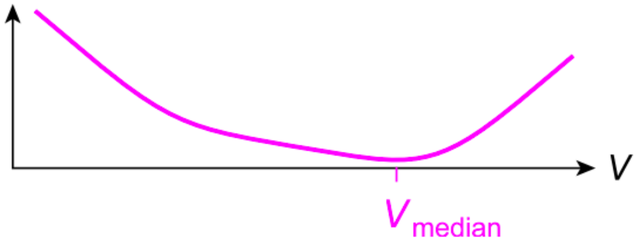
$$\sum_n \delta_n^2$$

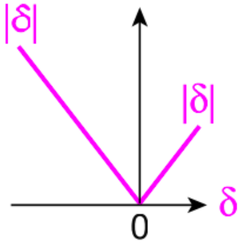




(C) Mean absolute error

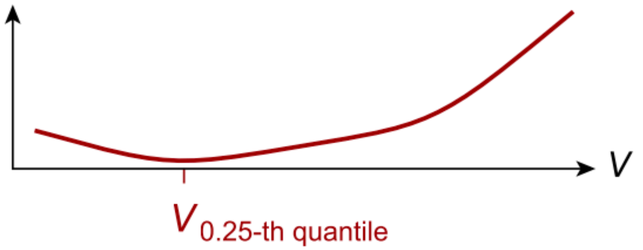
$$\sum_n |\delta_n|$$

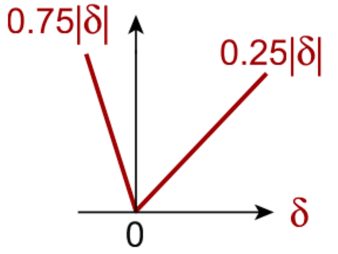




(D) Quantile regression loss

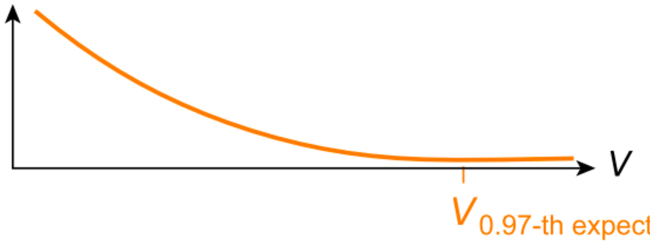
$$\sum_n |\delta_n| \cdot \begin{cases} 0.75 & \text{if } \delta_n \le 0 \\ 0.25 & \text{if } \delta_n > 0 \end{cases}$$

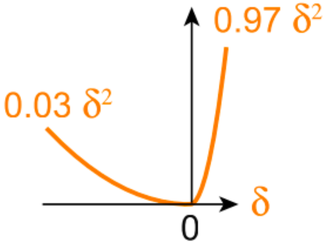


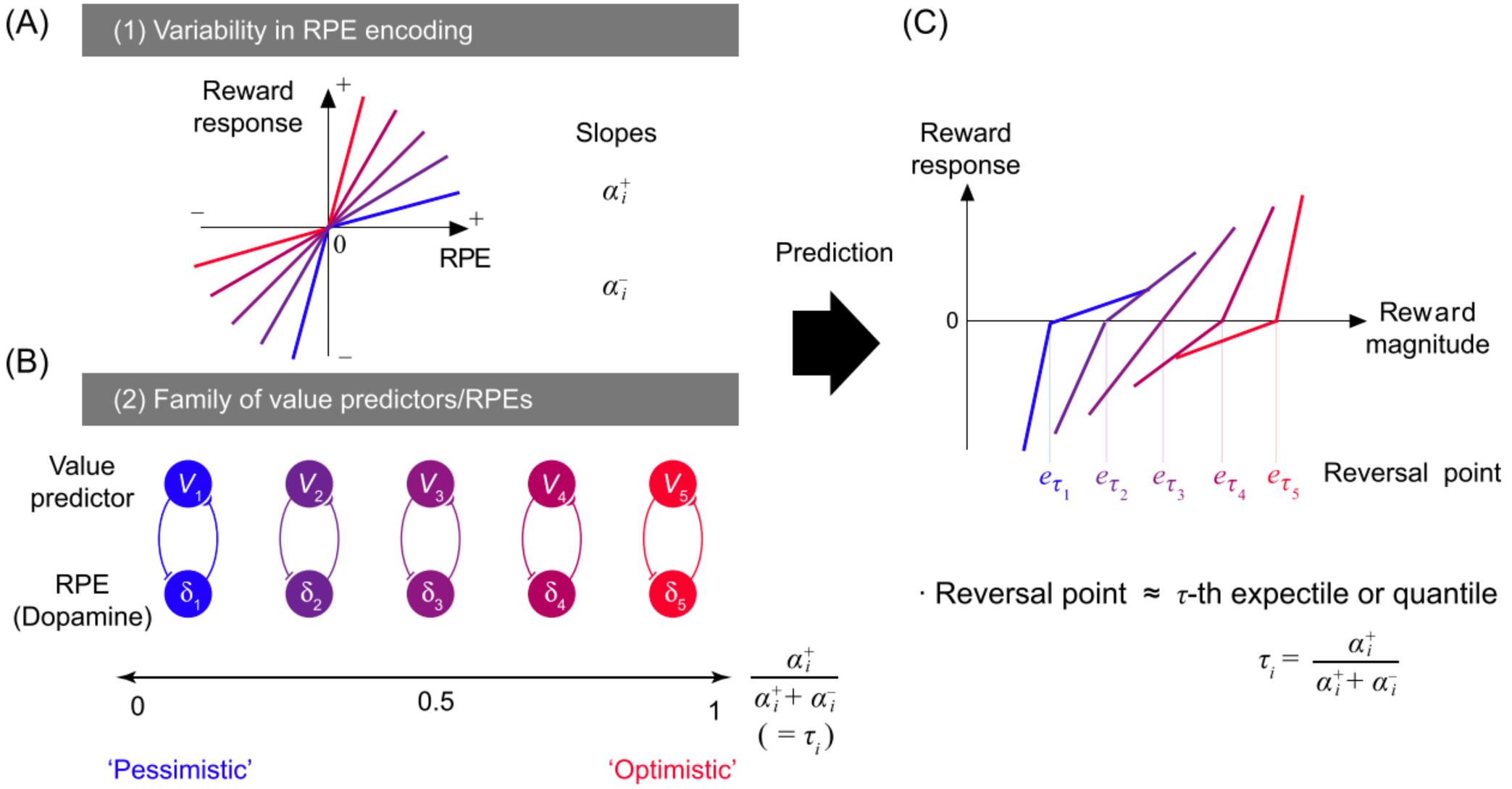


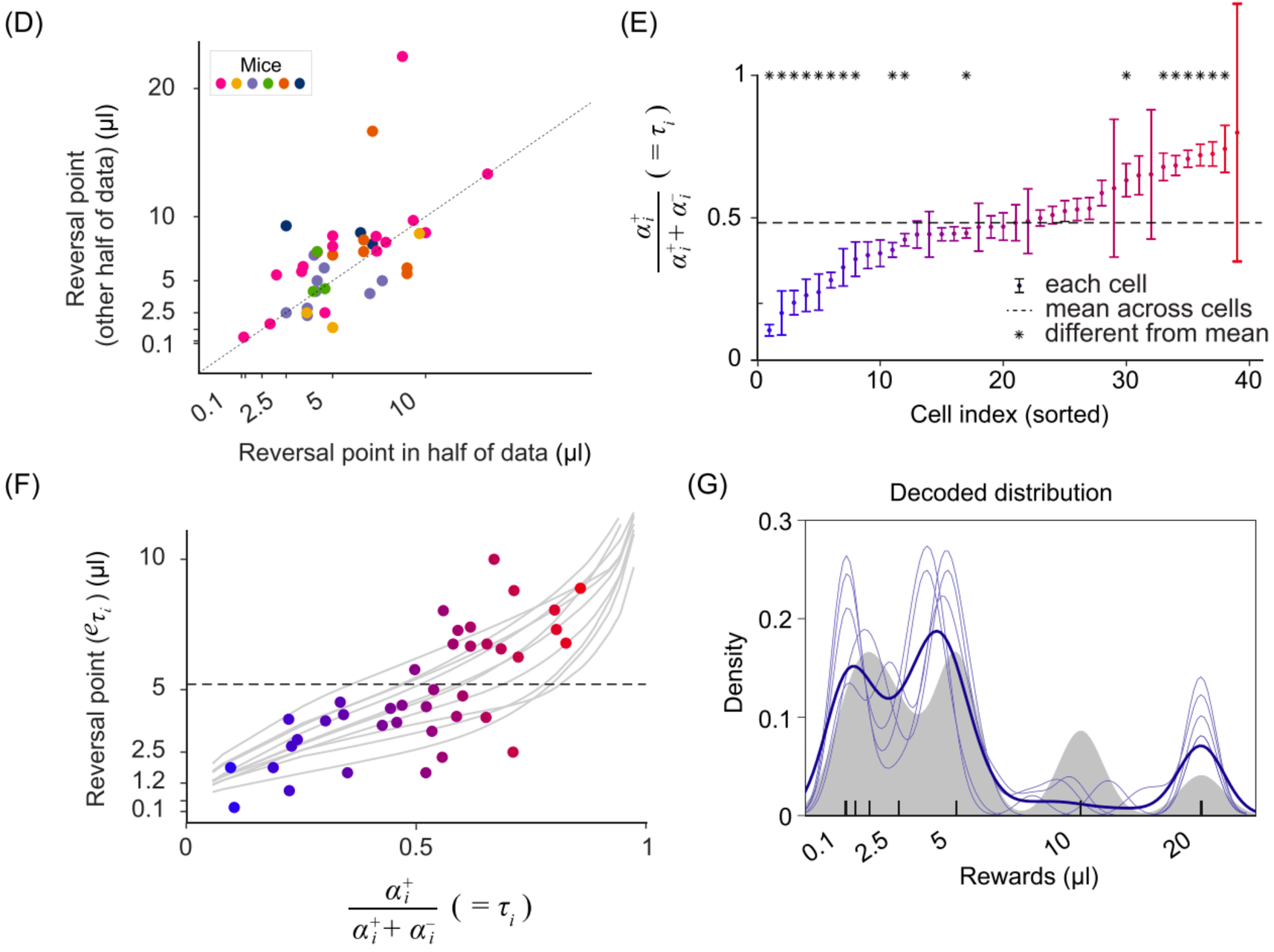
(E) Expectile regression loss

$$\sum_n \delta_n^2 \cdot \begin{cases} 0.03 & \text{if } \delta_n \le 0 \\ 0.97 & \text{if } \delta_n > 0 \end{cases}$$






(A)
(1) Variability in RPE encoding
Reward response
+
–
+
0
RPE
–
Slopes
α_i^+
α_i^-
(B)
(2) Family of value predictors/RPEs
Value predictor
RPE (Dopamine)
V_1 V_2 V_3 V_4 V_5
δ_1 δ_2 δ_3 δ_4 δ_5
0
0.5
1
$\frac{\alpha_i^+}{\alpha_i^+ + \alpha_i^-}$ $(=\tau_i)$
'Pessimistic'
'Optimistic'
Prediction
(C)
Reward response
0
Reward magnitude
e_{τ_1} e_{τ_2} e_{τ_3} e_{τ_4} e_{τ_5}
Reversal point
· Reversal point ≈ τ-th expectile or quantile
$\tau_i = \frac{\alpha_i^+}{\alpha_i^+ + \alpha_i^-}$


(D)
Mice
Reversal point (other half of data) (μl)
20
10
5
2.5
0.1
0.1
2.5
5
10
Reversal point in half of data (μl)
(E)
1
0.5
0
0
10
20
30
40
Cell index (sorted)
each cell
mean across cells
different from mean
(F)
Reversal point (e_{τ_i}) (μl)
10
5
2.5
1.2
0.1
0
0.5
1
$\frac{\alpha_i^+}{\alpha_i^+ + \alpha_i^-}$ ($= \tau_i$)
(G)
Decoded distribution
Density
0.3
0.2
0.1
0
0.1
2.5
5
10
20
Rewards (μl)

Distributional TD learning

- In distributional TD learning, one needs to sample the value function from the distribution
- It is easy in the quantile code by taking linear combination
- Currently, the performance boost from distributional RL comes from its benefit on learning the representation. How does knowing a distribution involves in decision making is another question.