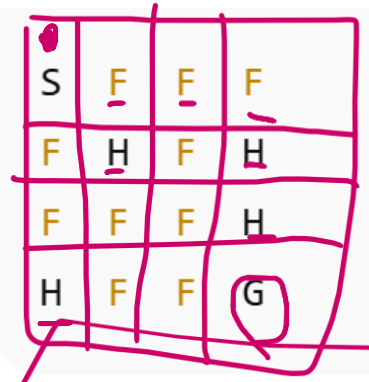
Reinforcement Learning Process

1. **Initialize** — Agent starts with no knowledge or policy.
2. **Explore** — Tries random actions to gather experience and rewards.
3. **Learn** — Updates value estimates or policy based on feedback.
4. **Improve** — Refines the policy to favor rewarding actions.
5. **Exploit** — Follows the learned policy to maximize rewards.

Frozen Lake

4x4

grid



- **S** = Start (state 0)
- **F** = Frozen (safe)
- **H** = Hole (terminal 0 reward)
- **G** = Goal (terminal +1 reward)

r
L
u
d

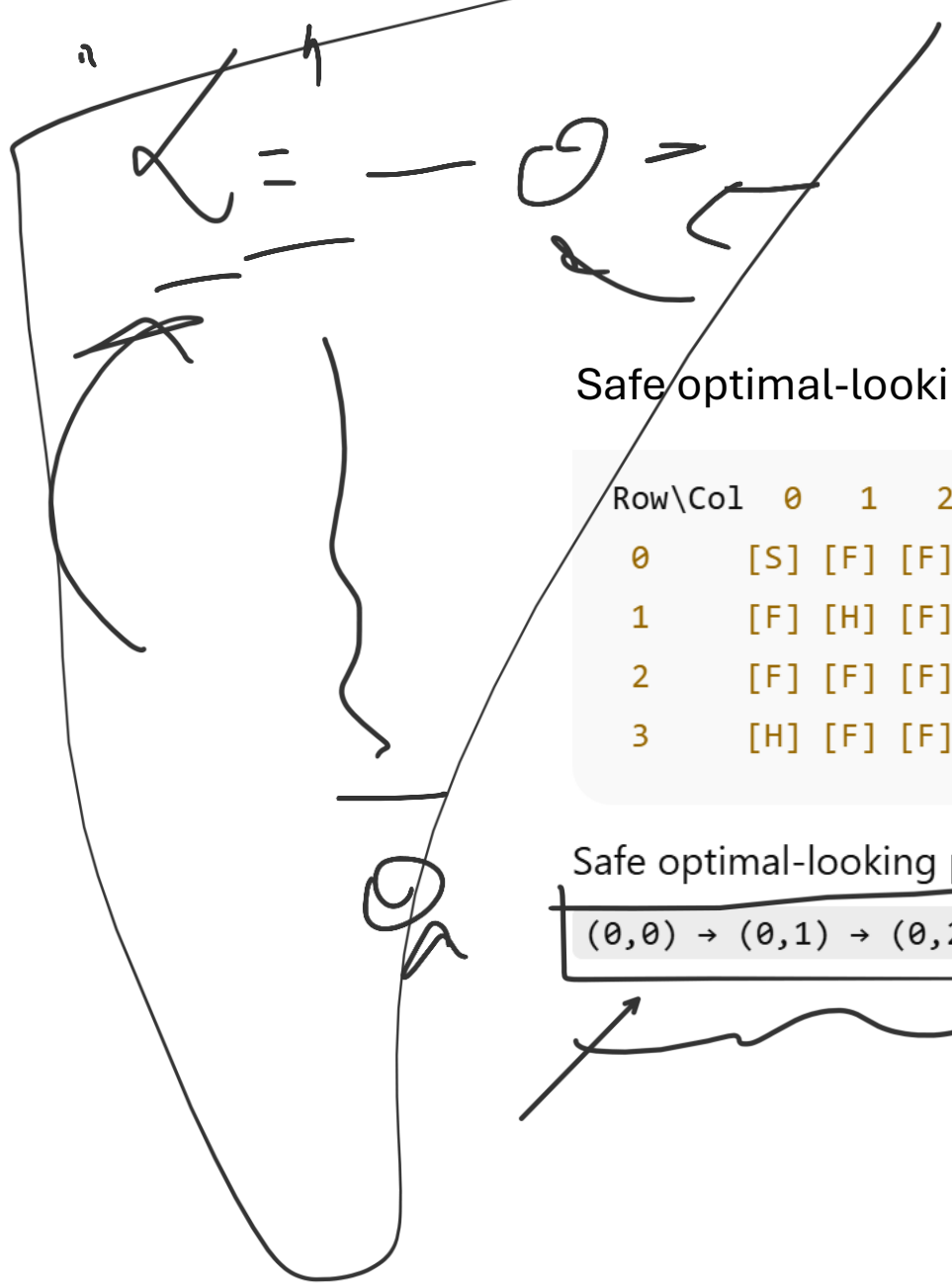
Initial State-Value Grid $V(s) = 0$

```
Initial State Values (Episode 1)
0.00   0.00   0.00   0.00
0.00     H    0.00     H
0.00   0.00   0.00     H
  H    0.00   0.00   1.00
```

Initial Policy Grid

```
Initial Policy (Episode 1)
?   ?   ?   ?
?   H   ?   H
?   ?   ?   H
H   ?   ?   G
```

- Each "?" means: agent may pick any of {←, ↓, →, ↑} with equal probability.
- Holes (`H`) and goal (`G`) are terminal — no actions there.



Safe optimal-looking path

```
Row\Col  0   1   2   3
 0     [S] [F] [F] [F]
 1     [F] [H] [F] [H]
 2     [F] [F] [F] [H]
 3     [H] [F] [F] [G]
```

Safe optimal-looking path we want the agent to discover (one valid route):

`(0,0) → (0,1) → (0,2) → (1,2) → (2,2) → (3,2) → (3,3)`

Episode 1

```
0.00  0.00  0.00  0.00
0.00   H    0.00   H
0.00  0.00  0.00   H
 H    0.00  0.00  1.00
```

```
Row\Col  0    1    2    3
 0      [S]  [F]  [F]  [F]
 1      [F]  [H]  [F]  [H]
 2      [F]  [F]  [F]  [H]
 3      [H]  [F]  [F]  [G]
```

Path Taken

(0,0) → (0,1) → (1,1) ❌ (hole → episode ends)

Updated State-Value Grid (after episode)

No positive reward was received (agent fell into a hole),
so values remain unchanged.

```
0.00  0.00  0.00  0.00
0.00   H    0.00   H
0.00  0.00  0.00   H
 H    0.00  0.00  1.00
```

Policy Grid (after episode)

All Q-values ≈ 0 → argmax ties → arbitrary (← chosen).
So, the current greedy policy is:

```
←  ←  ←  ←
←  H  ←  H
←  ←  ←  H
H  ←  ←  G
```

Episode 2

```
0.00  0.00  0.00  0.00
0.00   H    0.00   H
0.00  0.00  0.00   H
 H    0.00  0.00  1.00
```

```
Row\Col  0    1    2    3
 0      [S]  [F]  [F]  [F]
 1      [F]  [H]  [F]  [H]
 2      [F]  [F]  [F]  [H]
 3      [H]  [F]  [F]  [G]
```

Path Taken

(0,0) → (0,1) → (0,2) → (1,2) → (2,2) → (3,2) → (3,3) ✅ (goal)

Updated State-Value Grid (after episode)

```
0.30  0.45  0.55  0.10
0.25   H    0.65   H
0.20  0.40  0.80   H
 H    0.55  0.90  1.00
```

Policy Grid (after episode)

```
→  →  ↓  ←
↓  H  ↓  H
→  →  ↓  H
H  →  →  G
```

Values increase closer to the goal because those states lead reliably to the +1 reward.

Episode 3

(Carries over from Episode 2 — the agent already knows a good path.)

0.30	0.45	0.55	0.10
0.25	H	0.65	H
0.20	0.40	0.80	H
H	0.55	0.90	1.00

Row\Col	0	1	2	3
0	[S]	[F]	[F]	[F]
1	[F]	[H]	[F]	[H]
2	[F]	[F]	[F]	[H]
3	[H]	[F]	[F]	[G]

Path Taken

(0,0) → (0,1) → (0,2) → (1,2) → (2,2) → (3,2) → (3,3) ✅ (goal)

Updated State-Value Grid (after episode)

0.45	0.60	0.70	0.10
0.40	H	0.75	H
0.35	0.55	0.85	H
H	0.70	0.92	1.00

Policy Grid (after episode)

→	→	↓	←
↓	H	↓	H
→	→	↓	H
H	→	→	G

Episode 4

0.45	0.60	0.70	0.10
0.40	H	0.75	H
0.35	0.55	0.85	H
H	0.70	0.92	1.00

Row\Col	0	1	2	3
0	[S]	[F]	[F]	[F]
1	[F]	[H]	[F]	[H]
2	[F]	[F]	[F]	[H]
3	[H]	[F]	[F]	[G]

Path Taken

Agent mostly exploits but explores a small detour then returns to the known path

(0,0) → (0,1) → (1,1) ❌ (hole) (explore)

(restart) (0,0) → (0,1) → (0,2) → (1,2) → (2,2) → (3,2) → (3,3) ✅

Updated State-Value Grid (after episode)

0.52	0.66	0.76	0.12
0.48	H	0.80	H
0.40	0.62	0.88	H
H	0.75	0.94	1.00

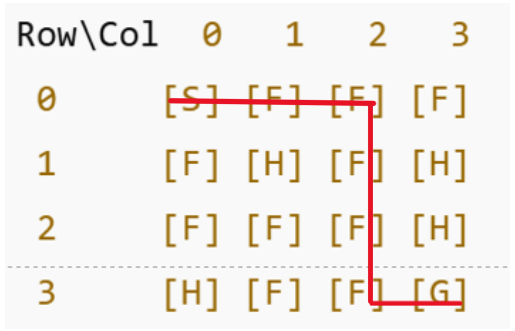
Policy Grid (after episode)

→	→	↓	←
↓	H	↓	H
→	→	↓	H
H	→	→	G

the successful run reinforces the main path; the hole visit produces no positive reward but teaches the agent to avoid (lowers relative preference).

Episode 5

```
0.52  0.66  0.76  0.12
0.48   H    0.80   H
0.40  0.62  0.88   H
  H   0.75  0.94  1.00
```



Path Taken

Agent exploits the learned route deterministically

(0,0) → (0,1) → (0,2) → (1,2) → (2,2) → (3,2) → (3,3) ✅

Updated State-Value Grid (after episode)

```
0.60  0.70  0.80  0.13
0.56   H    0.83   H
0.50  0.68  0.90   H
  H   0.80  0.96  1.00
```

Policy Grid (after episode)

```
→  →  ↓  ←
↓  H  ↓  H
→  →  ↓  H
H  →  →  G
```

(Policy is now effectively deterministic along the optimal corridor.)

Here's how the **arrows (policy)** are derived from the **state values** $V(s)$ —the agent looks at the neighboring states' values and points toward the **highest one**.

State-Value Grid (V)

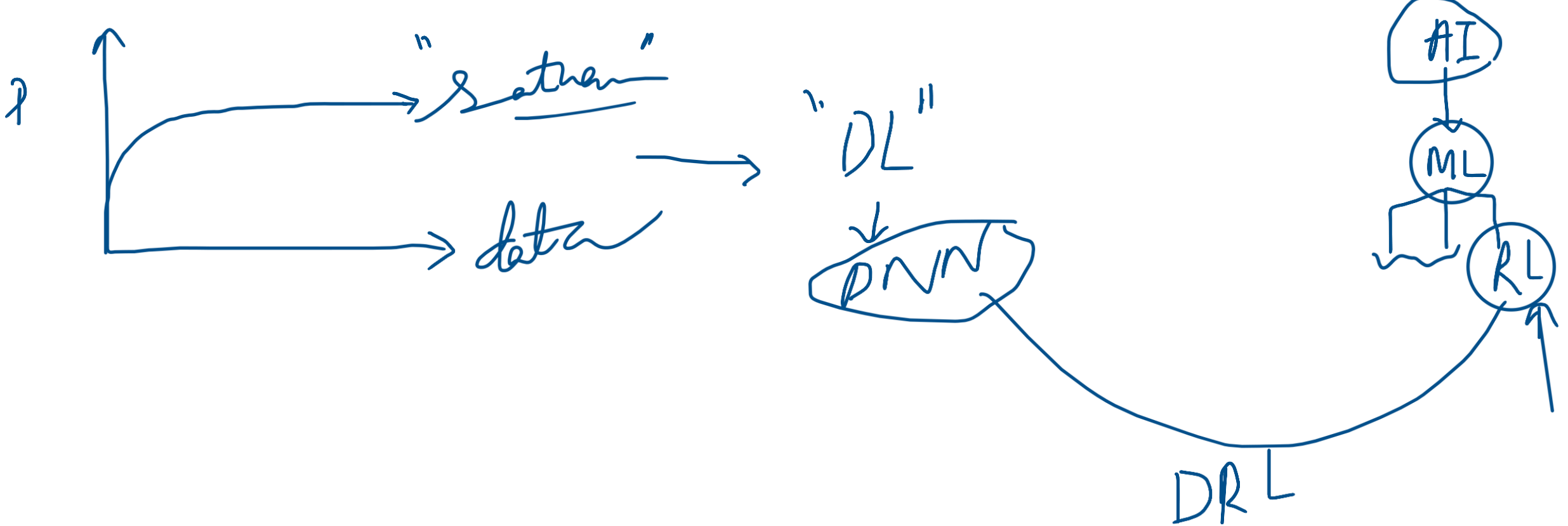
0.70	0.75	0.80	0.10
0.65	H	0.85	H
0.60	0.80	0.90	H
H	0.88	0.95	1.00

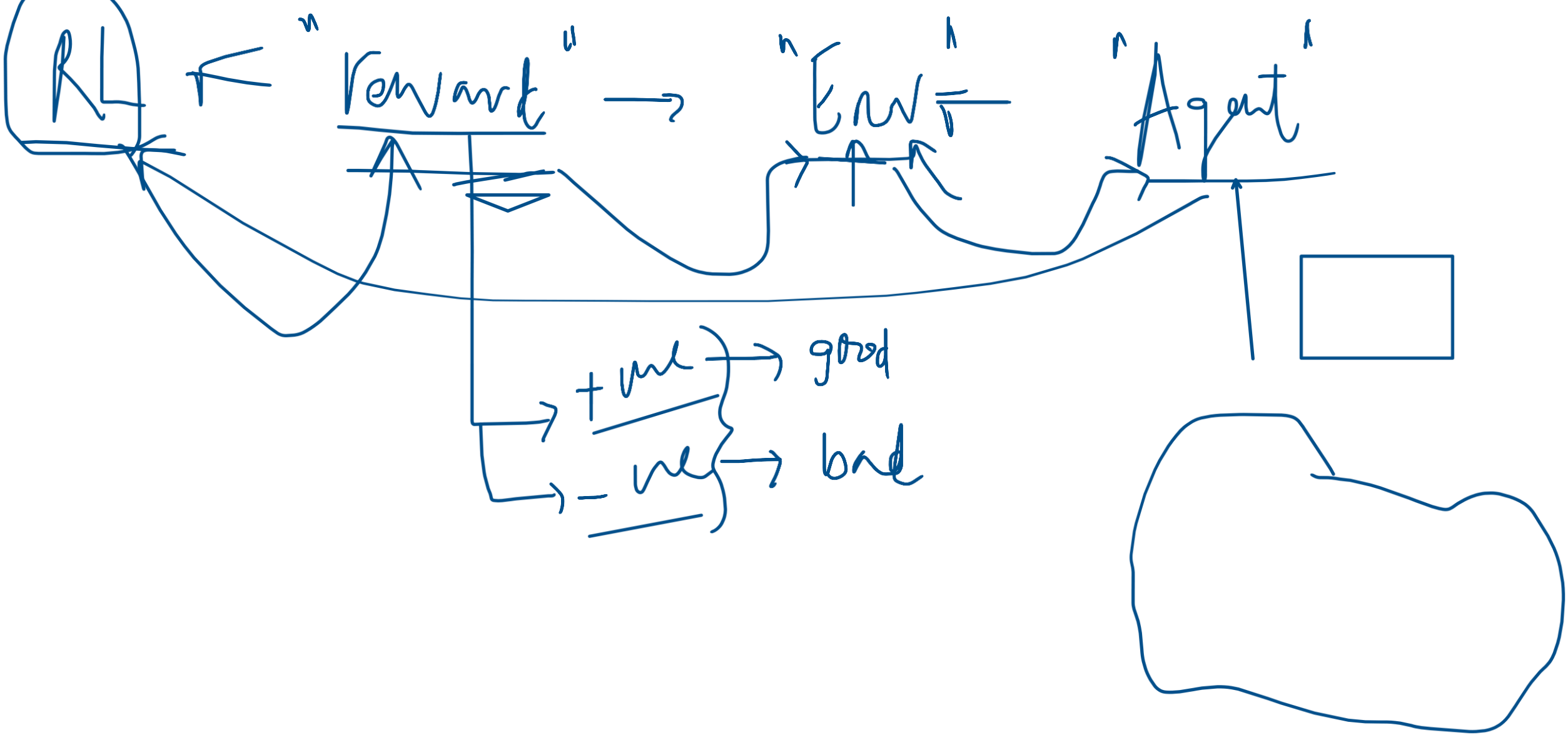
Derived Policy Grid

→	→	↓	←
↑	H	↓	H
→	→	↓	H
H	→	→	G

Interpretation:

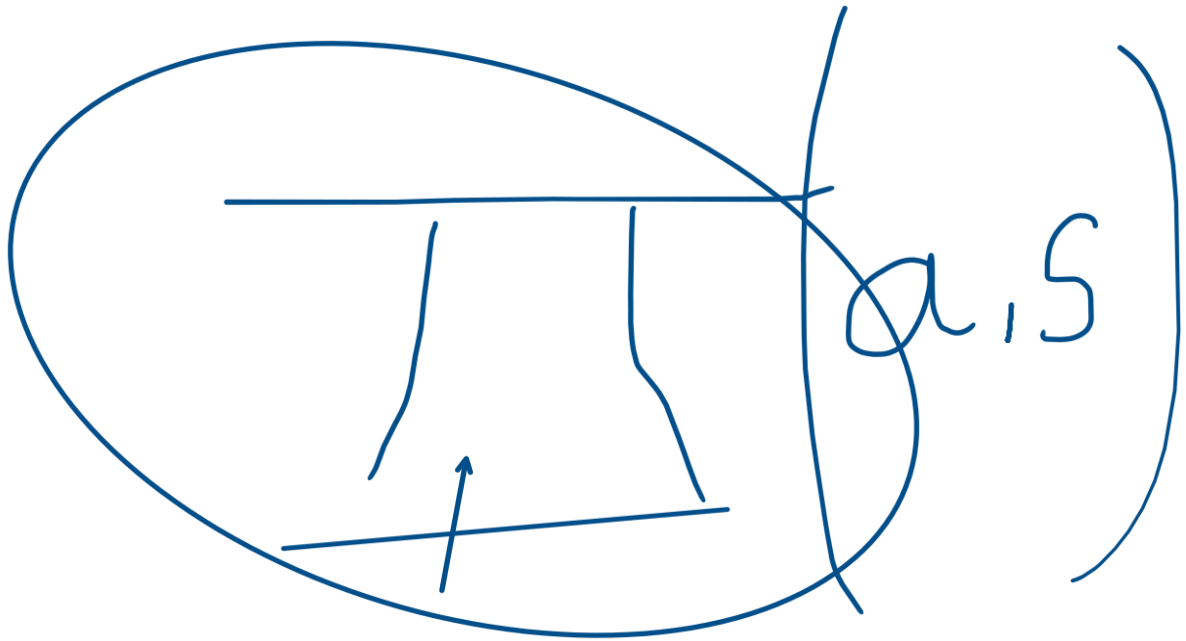
- Each arrow points toward the neighbor with the **highest next-state value**.
- For example:
  - At (0,0) value=0.70 → best neighbor is (0,1)=0.75 → arrow →
  - At (2,2) value=0.90 → best neighbor is (3,2)=0.95 → arrow ↓
- Holes (H) and Goal (G) are terminal — no arrows.


"Science"
data
"DL"
DNN
AI
ML
RL
DRL


RL
Reward
Env
Agent
+ve
good
-ve
bad

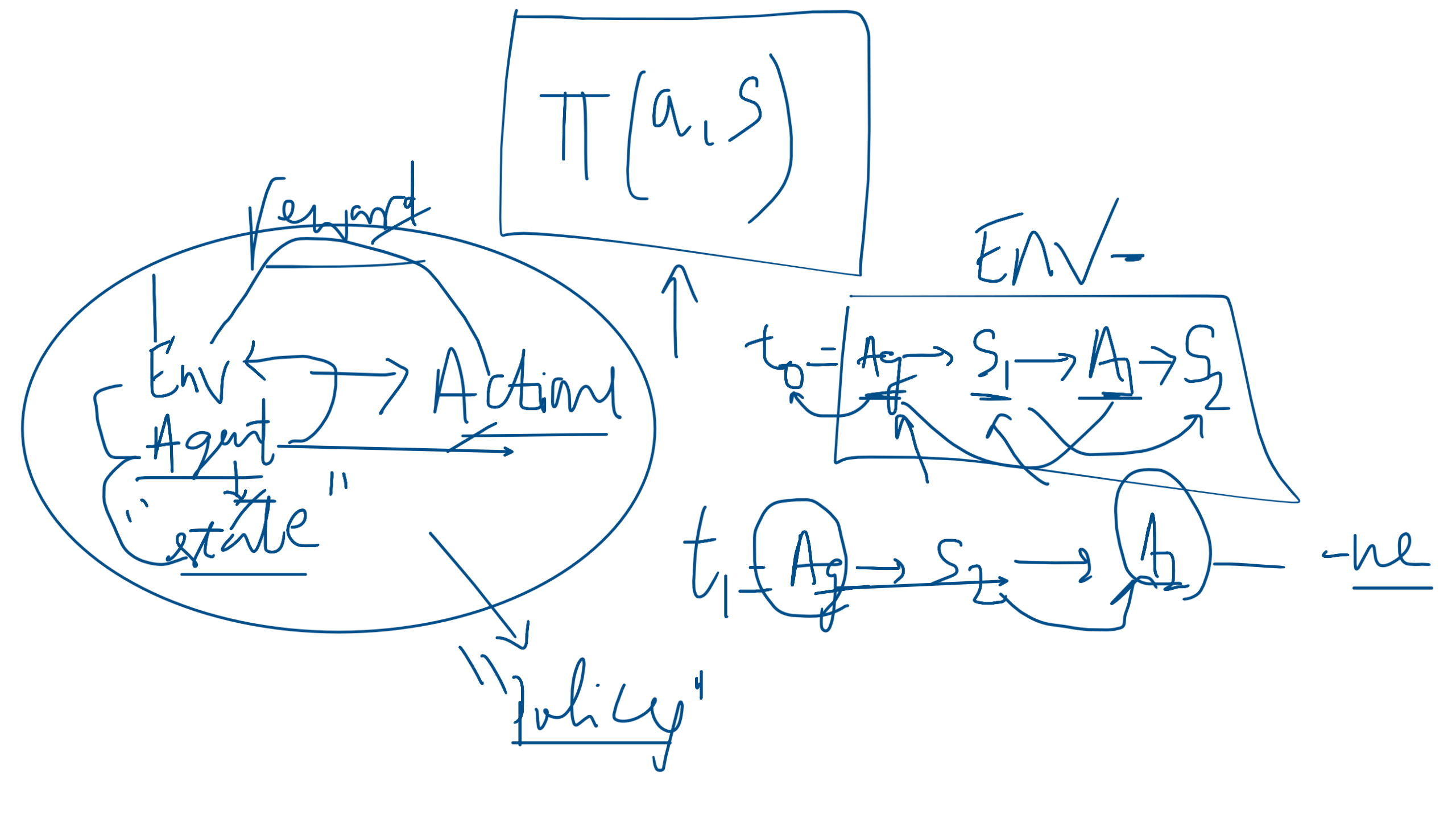
"Problem"

"π"

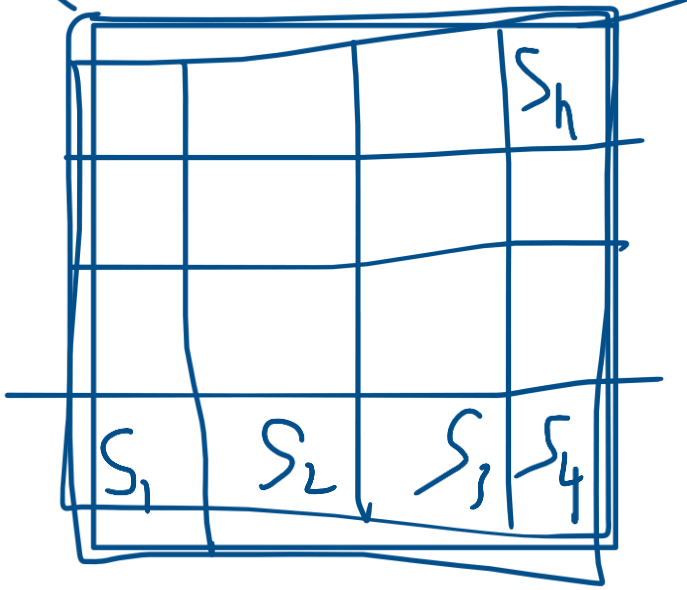


"acting" state

+ve reward


π(a, s)
Reward
Env
Agent
Actions
"state"
"policy"
ENV-
t_0 = Ag → S_1 → A_1 → S_2
t_1 = Ag → S_2 → A_2 —
-ne

Grid-world



ENV

Tic-Tac-Toe

P

OR

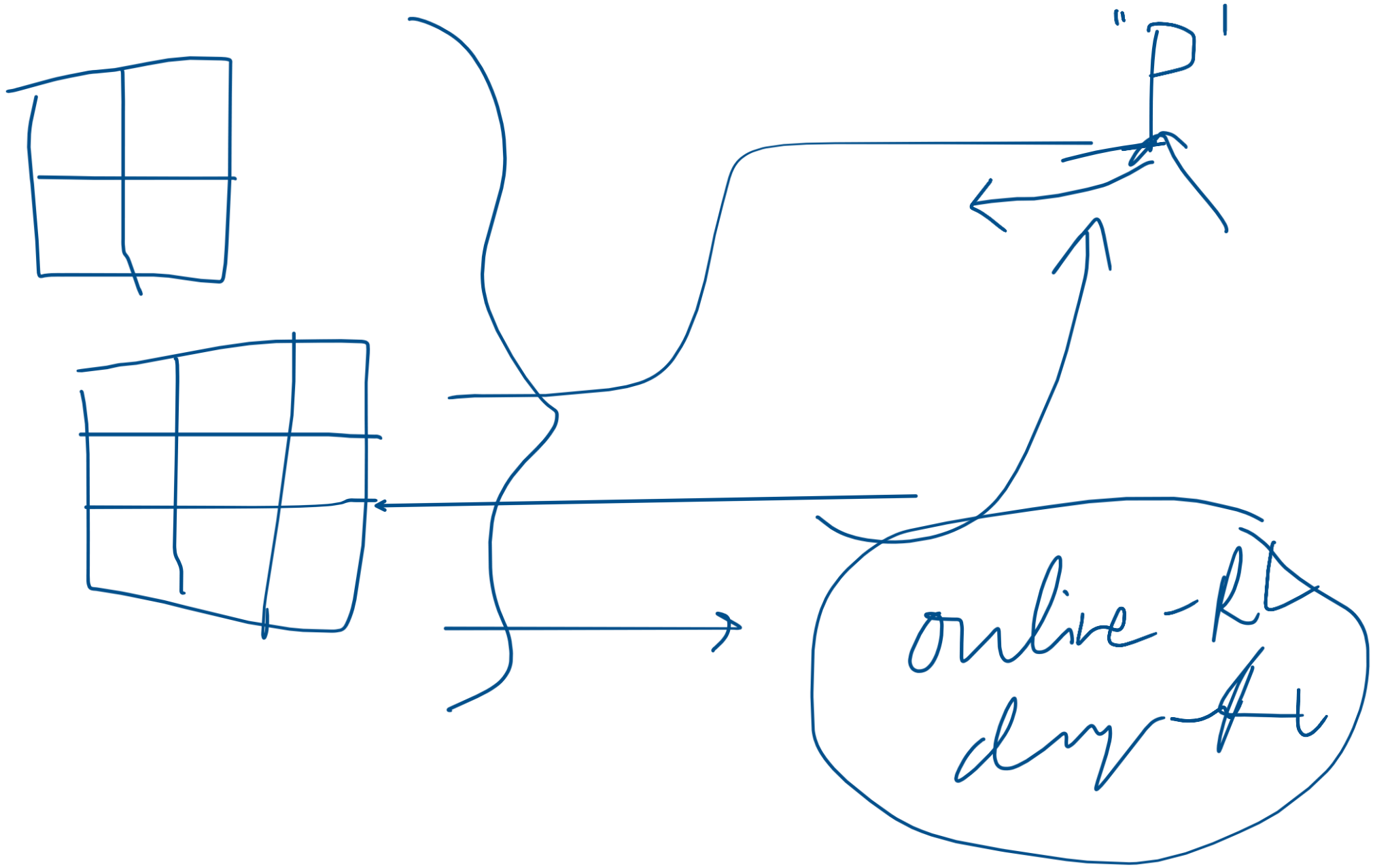
game

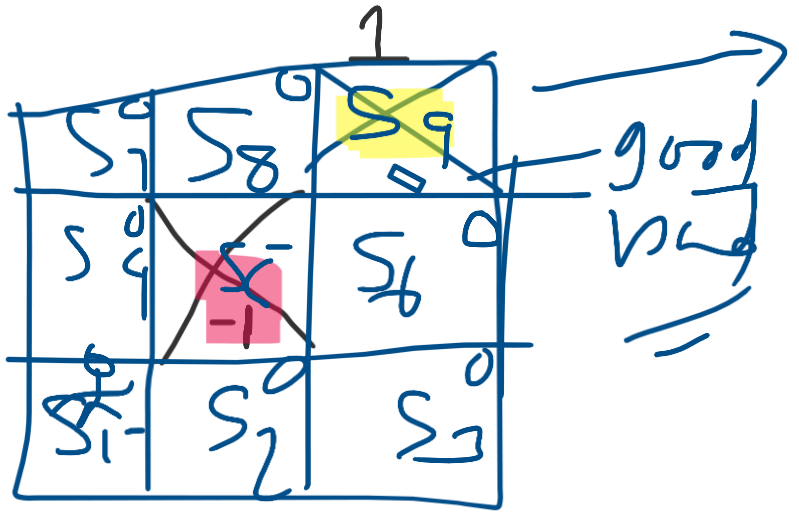
OR

navigation

OR

Mario!


"P"
online-RL
dyna

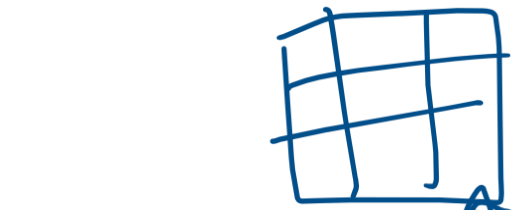


RL → ENV — 3×3 = n×n ✓

→ Agent — robot

→ States { $S_1 \rightarrow S_9$ }

→ Actions



$S_1 \rightarrow R, U, \cancel{L}, \cancel{D}$

$S_2 \rightarrow R, L, U$

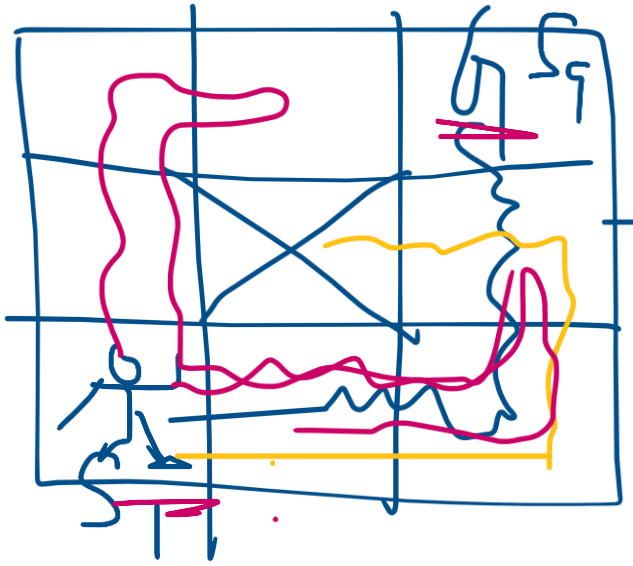
$S_5 \rightarrow R, L, U, D$ ✓

Forward

+ve → 1

-ve → -1

normal → 0



Initial

"Explore"

"Policy"

1) G ✓

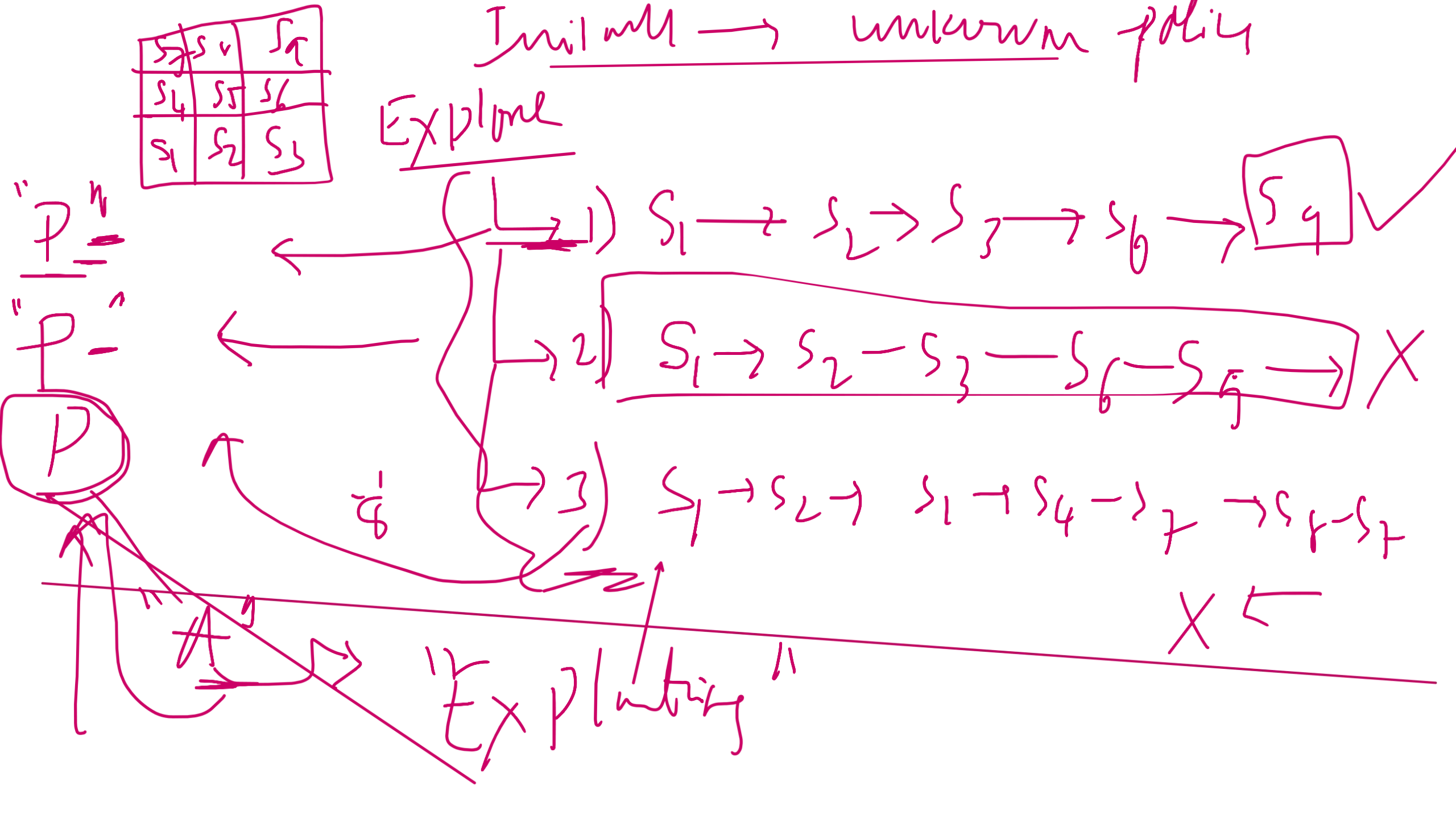
2) Trap / Bad → X ✓

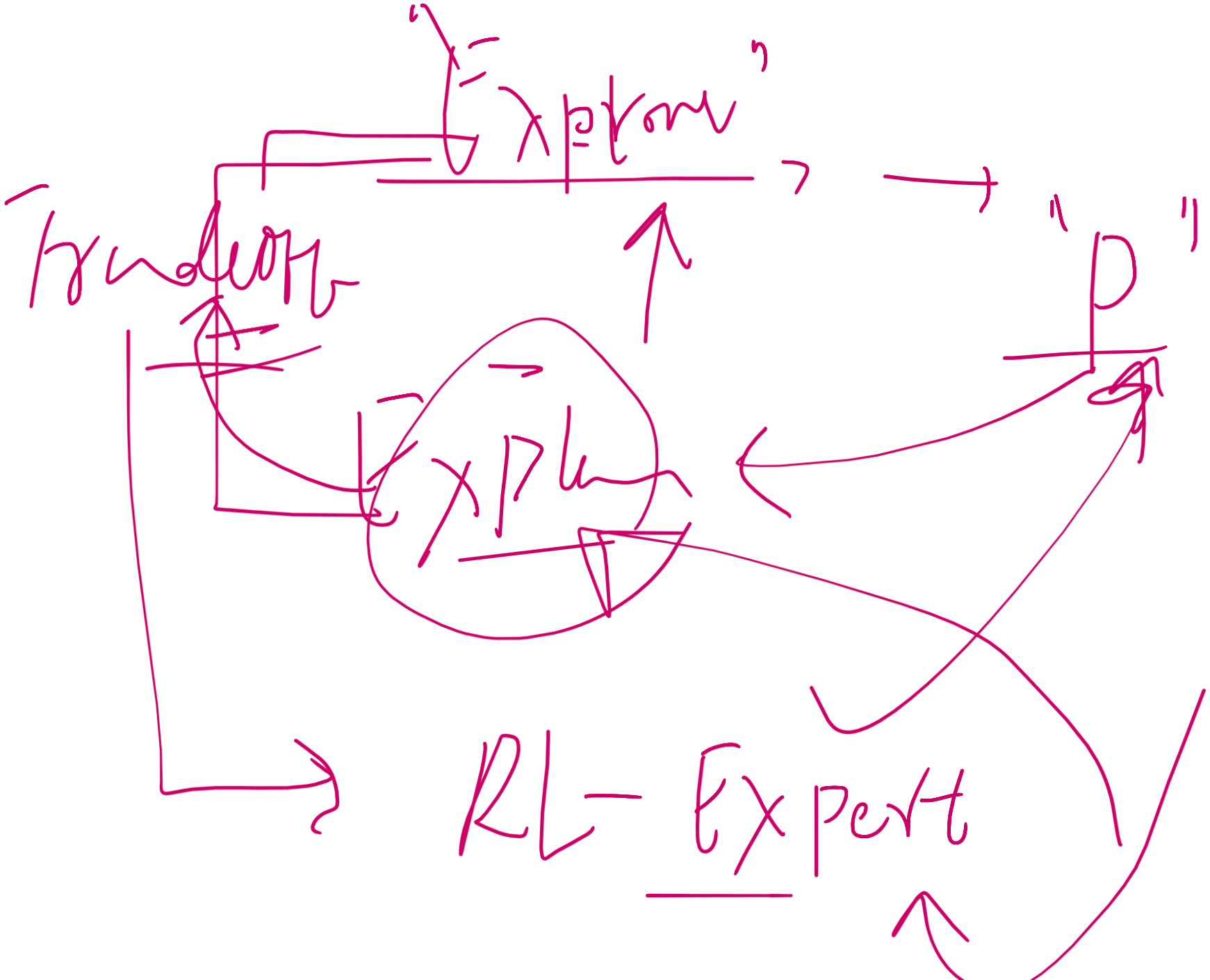
3) X ✓ "Constraint"

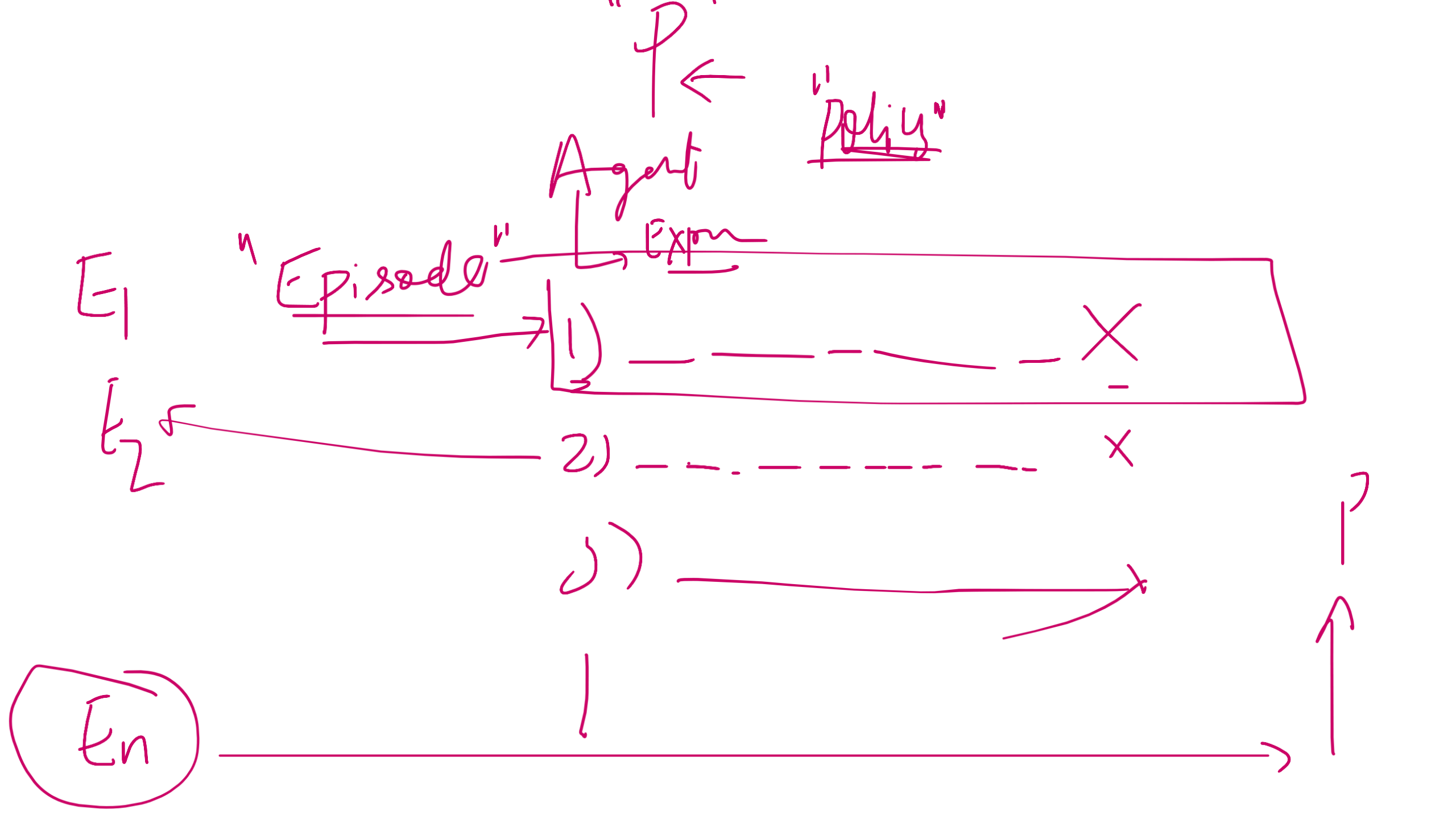
Timer

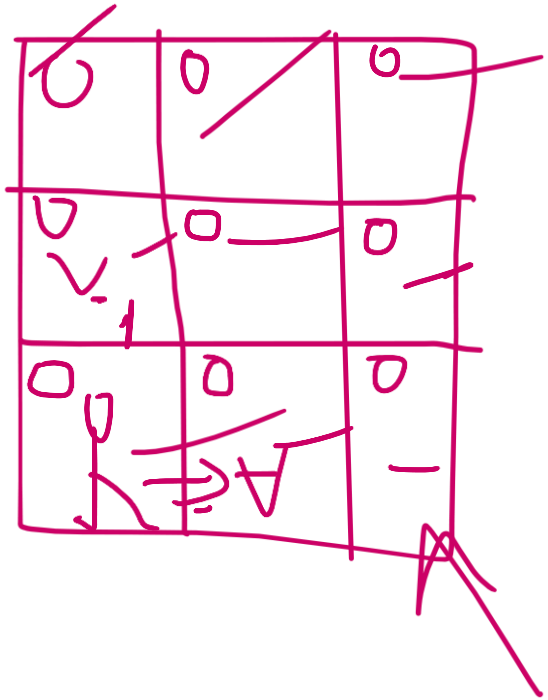
t

Step


Initial → uniform policy
Explore
"P"
"P-"
P
1) S1 → S2 → S3 → S6 → S9
2) S1 → S2 → S3 → S6 → S9 → X
3) S1 → S2 → S1 → S4 → S7 → S8 → S7
X
"Exploiting"


Random
Expert
"P"
Explore
RL- Expert


"P"
"Policy"
Agent
Exp
"Episode"
E1
1)
X
E2
2)
X
3)
En



RL → State Value → V(S)

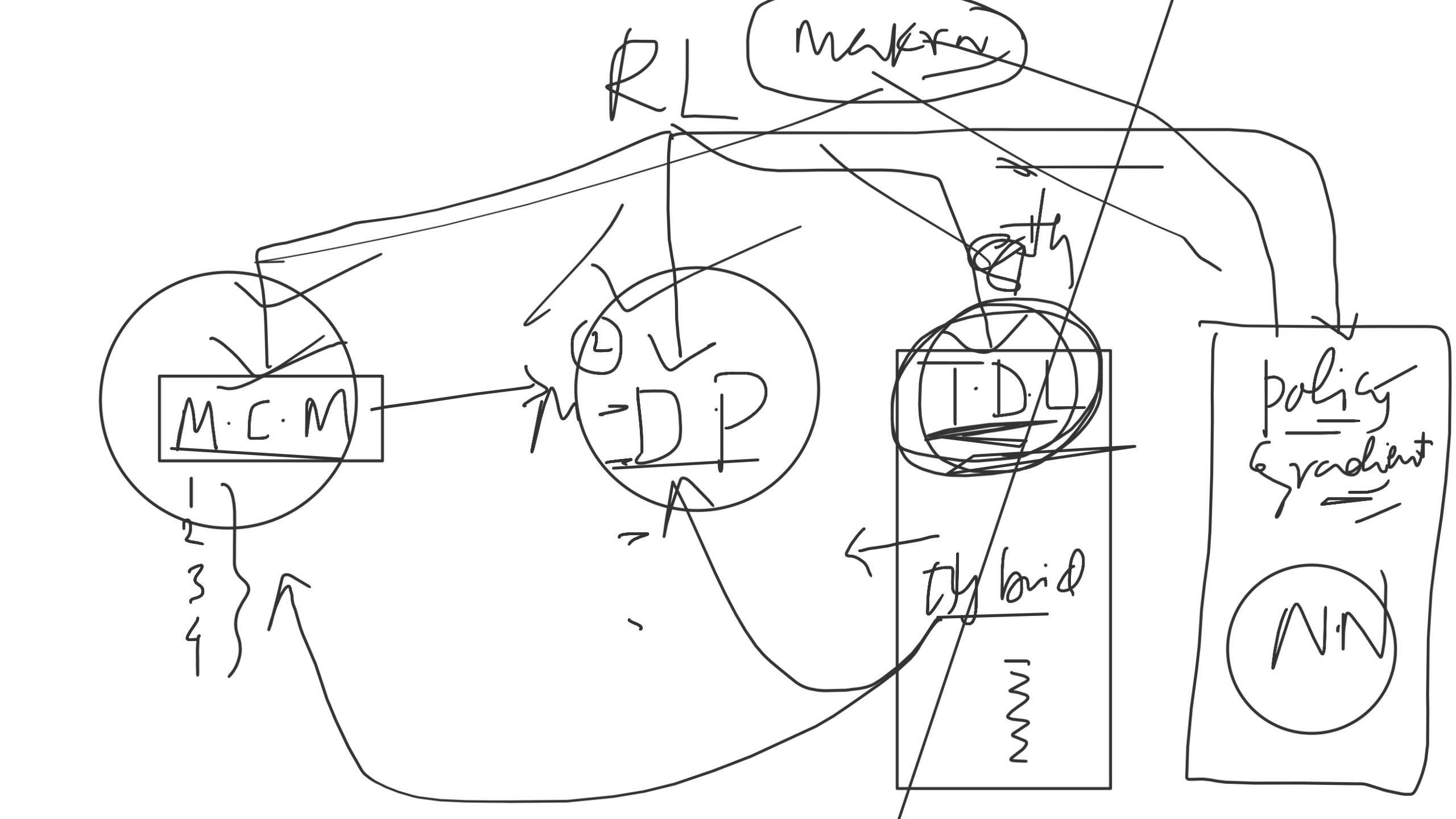
Action Value → Q(S,A)

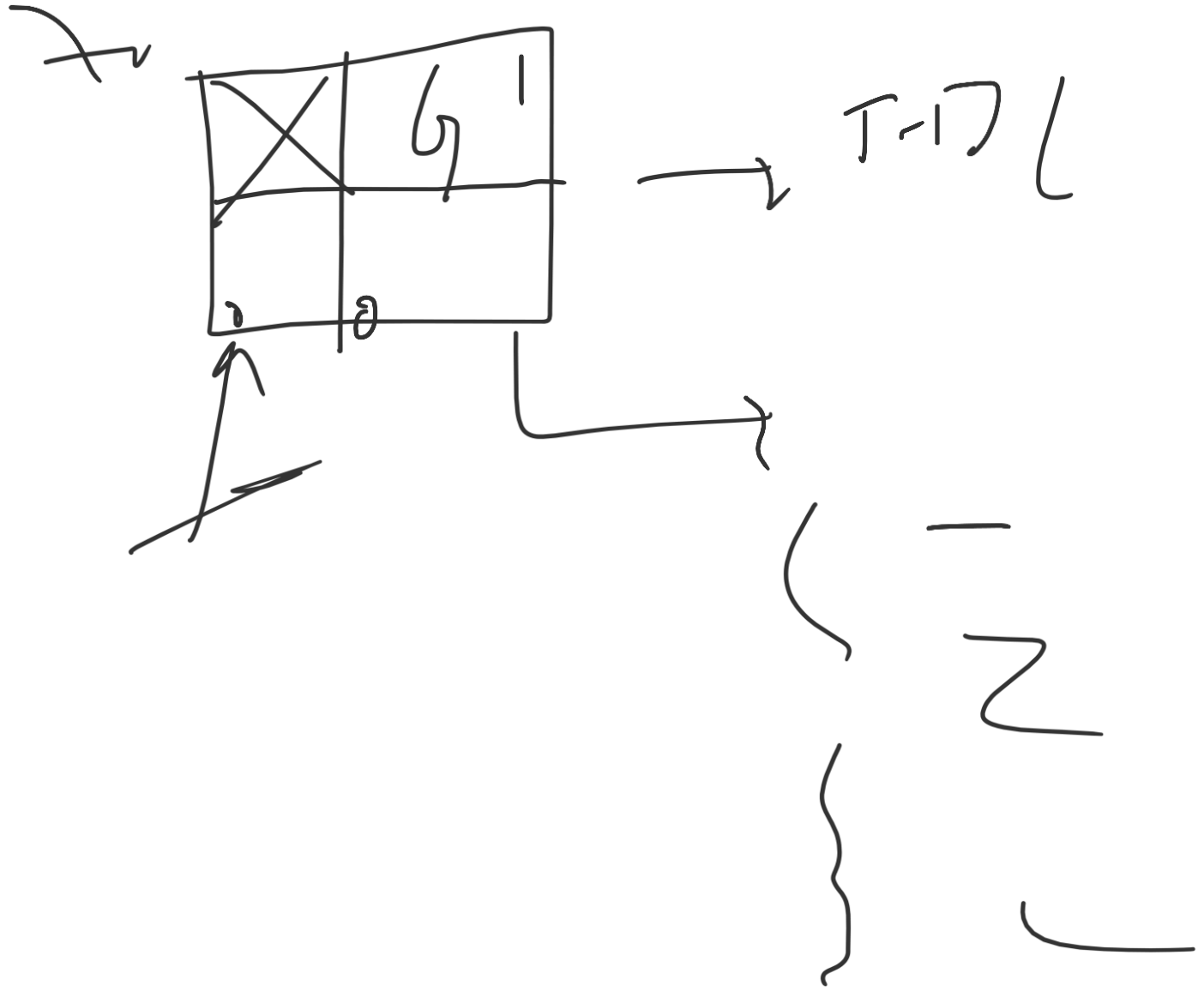
Initial each state "Value"

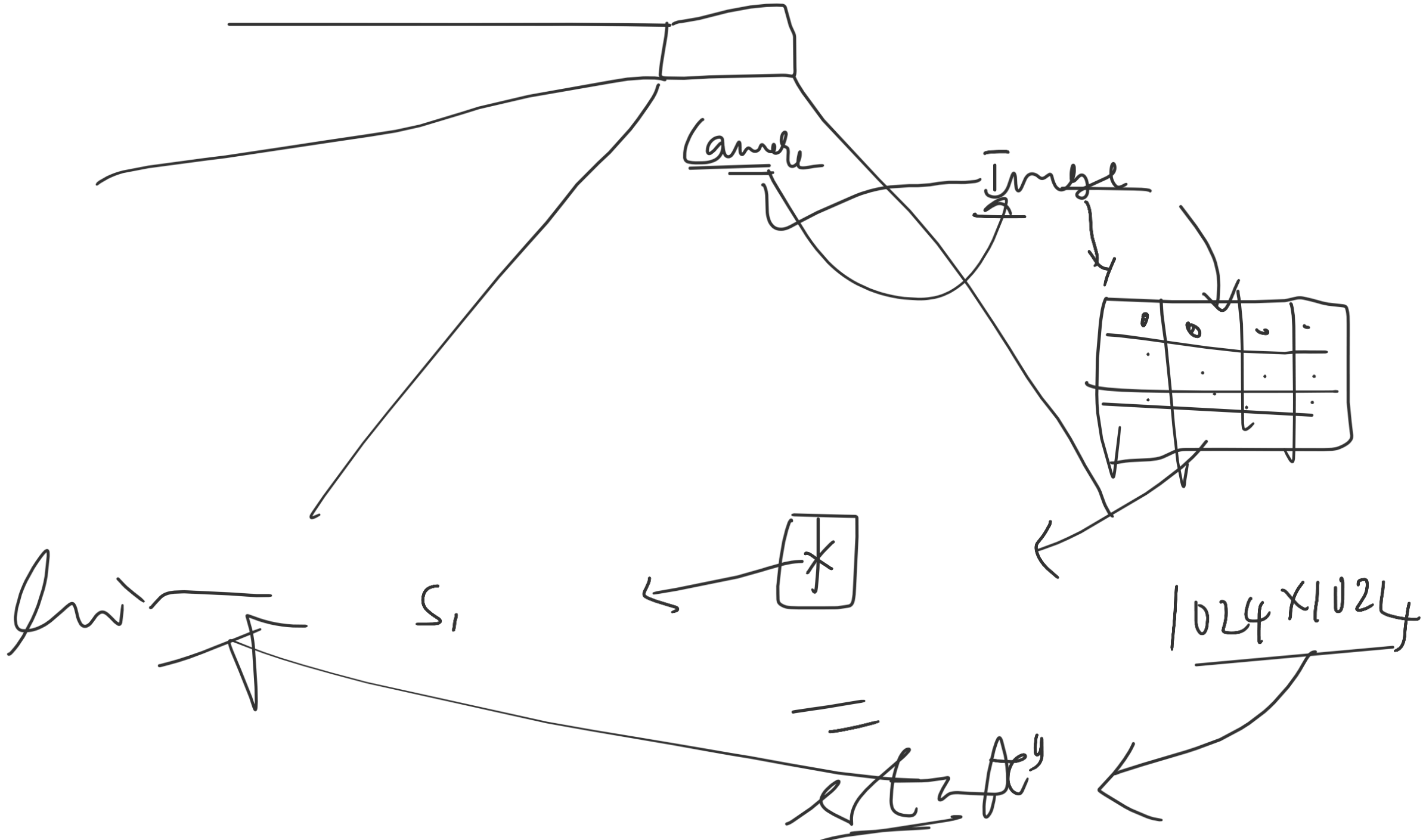
E_1 → Tech → Alg → Policy

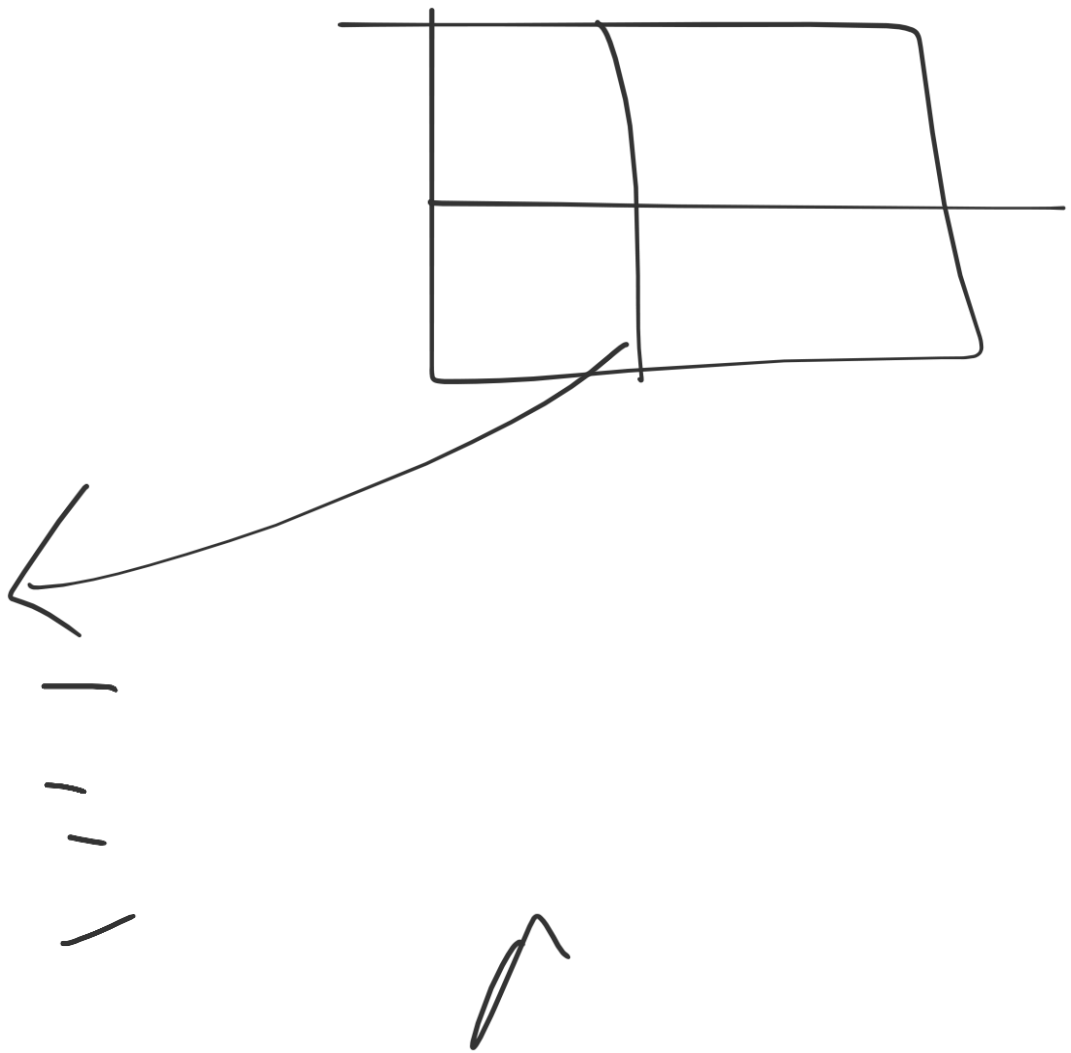
E_2 →

E

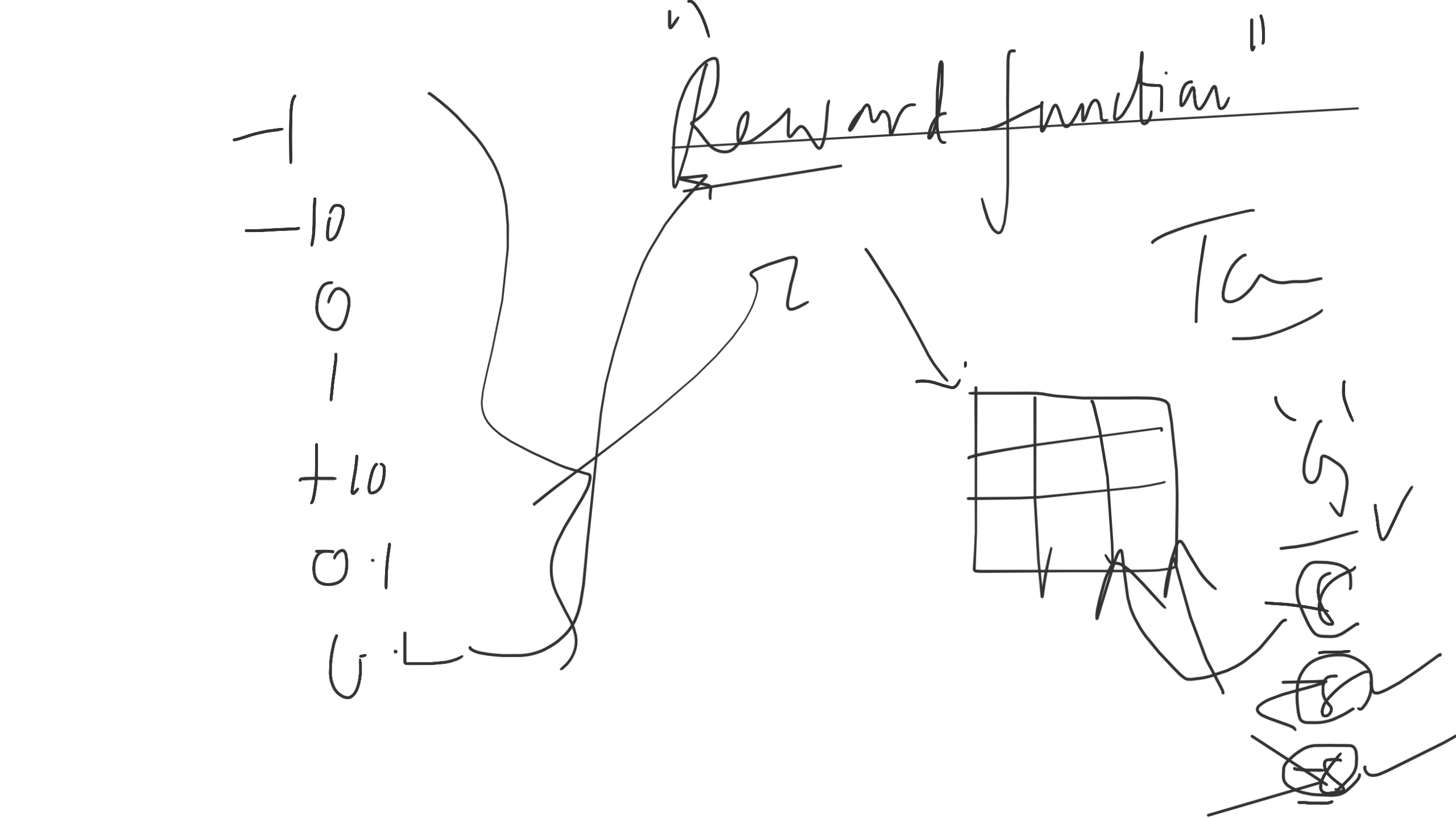

RL
Markov
M.C.M
M-D.P
T.D.L
Policy
Gradient
Hybrid
N.N




Camera
Image
1024 X 1024



$$V(s) + \alpha(s - 0)$$


Reward function
-1
-10
0
1
+10
0.1
0.1