OptionZero: Planning with Learned Options

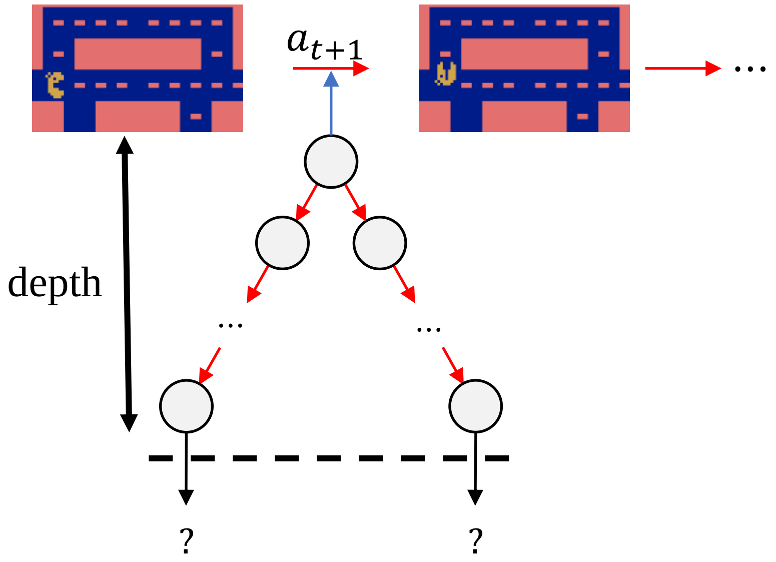
Po-Wei Huang[1,2], Pei-Chiun Peng[1,2], Hung Guei[1], Ti-Rong Wu[1]

[1] Academia Sinica
[2] National Yang Ming Chiao Tung University

Options in Planning

Planning with primitive actions

- Inefficient for straightforward paths
- Limited by the simulation budget

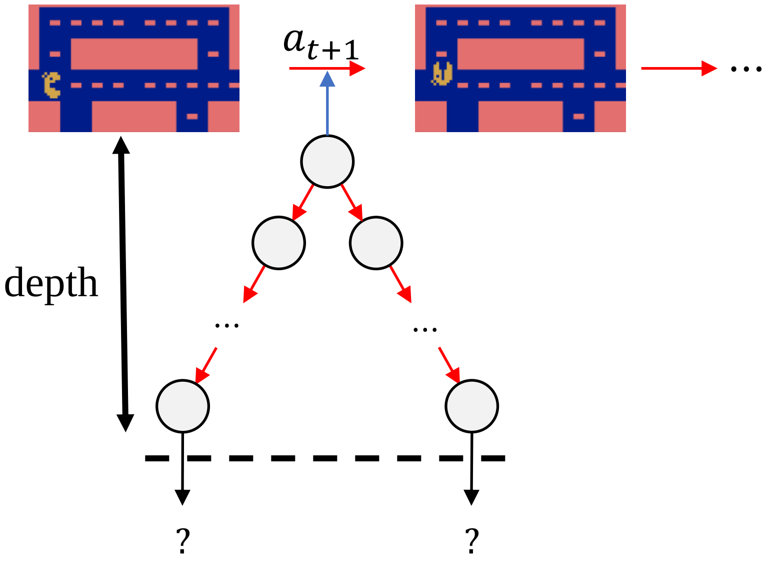


Options in Planning

option: temporary extended actions

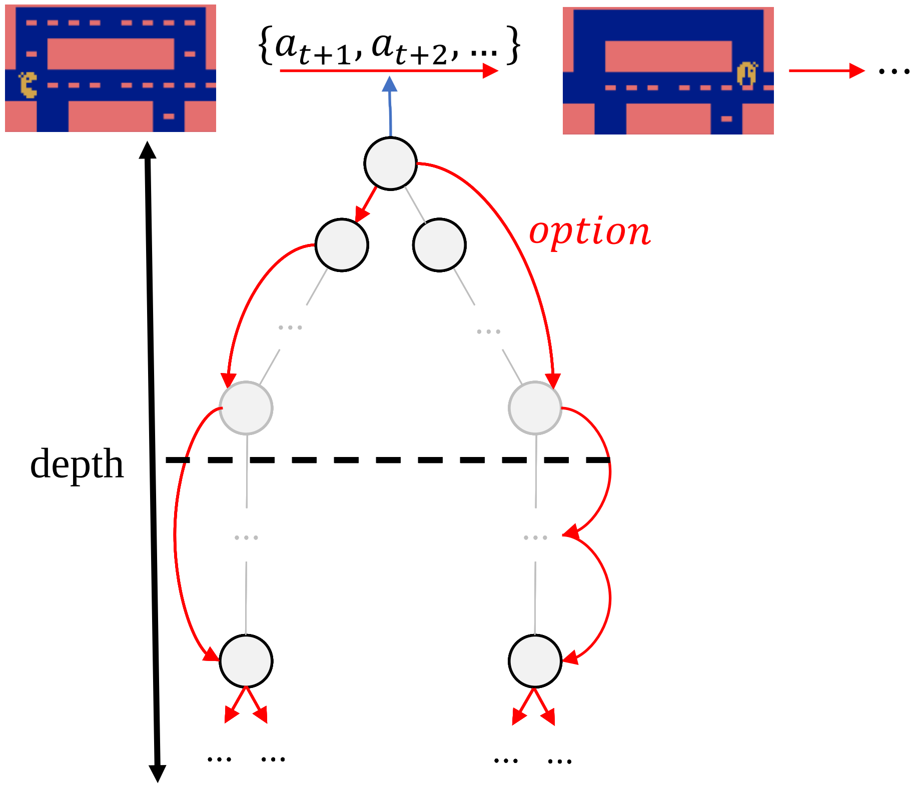
Planning with primitive actions

- Inefficient for straightforward paths
- Limited by the simulation budget



Planning with options

- Reduce the frequency of choices
- Search deeper under same budget



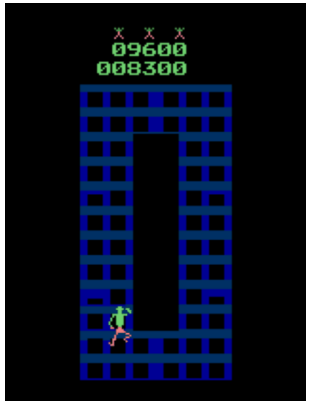
Motivation

- Limitations of previous works
  - Design of options: hard to generalize across different environments
    - Repeated primitive actions (Sharma et al., 2016; Durugkar et al., 2016; Lakshminarayanan et al. 2017)
    - Handcrafted options (Gabor et al., 2019; de Waard et al., 2016)
    - Learn from expert demonstration data (Czechowski et al., 2021; Kujanpää et al., 2023; 2024)

freeway

repeated **up**

crazy climber



interleaving **up** and **down**

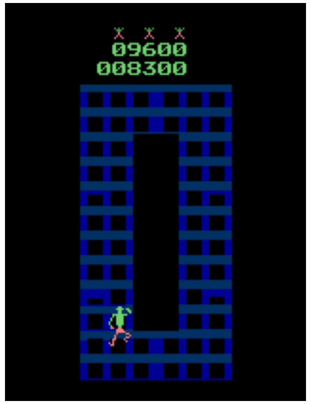
Motivation

- Limitations of previous works
  - Design of options: hard to generalize across different environments
    - Repeated primitive actions (Sharma et al., 2016; Durugkar et al., 2016; Lakshminarayanan et al. 2017)
    - Handcrafted options (Gabor et al., 2019; de Waard et al., 2016)
    - Learn from expert demonstration data (Czechowski et al., 2021; Kujanpää et al., 2023; 2024)
  - Planning with options: higher environment transition cost for options

freeway

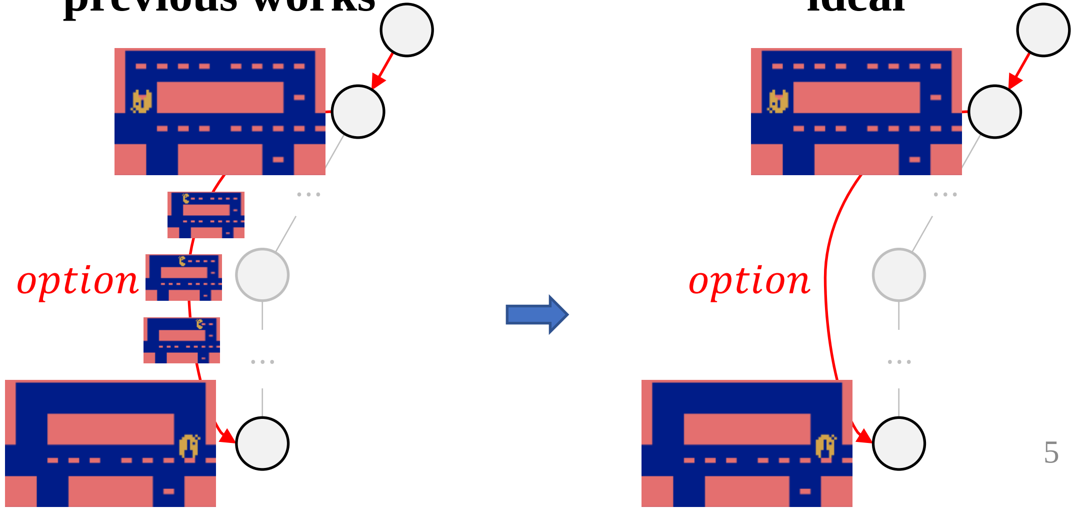
repeated **up**

crazy climber



interleaving **up** and **down**

previous works **ideal**



Motivation

- Limitations of previous works
  - Design of options: hard to generalize across different environments
    - Repeated primitive actions (Sharma et al., 2016; Durugkar et al., 2016; Lakshminarayanan et al. 2017)
    - Handcrafted options (Gabor et al., 2019; de Waard et al., 2016)
    - Learn from expert demonstration data (Czechowski et al., 2021; Kujanpää et al., 2023; 2024)
  - Planning with options: higher environment transition cost for options

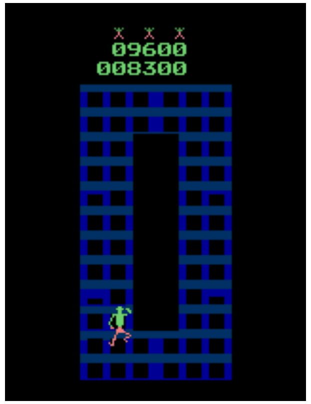
➔ We proposed **OptionZero**, which integrates options into **MuZero**

- Autonomously discover options through self-play games and utilize options during planning

freeway

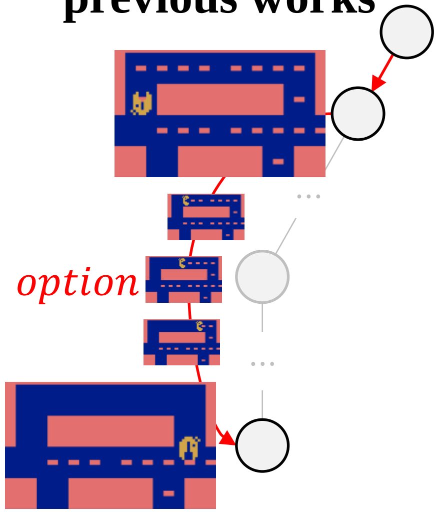
repeated **up**

crazy climber

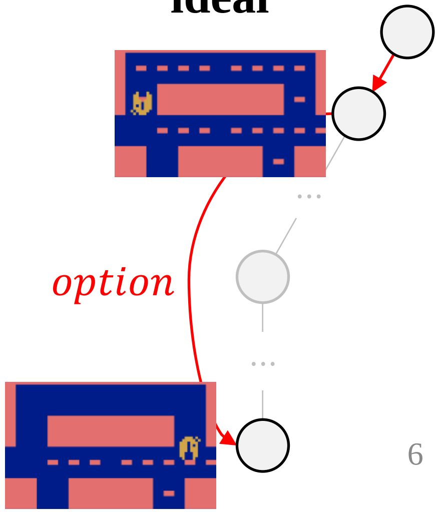


interleaving **up** and **down**

previous works

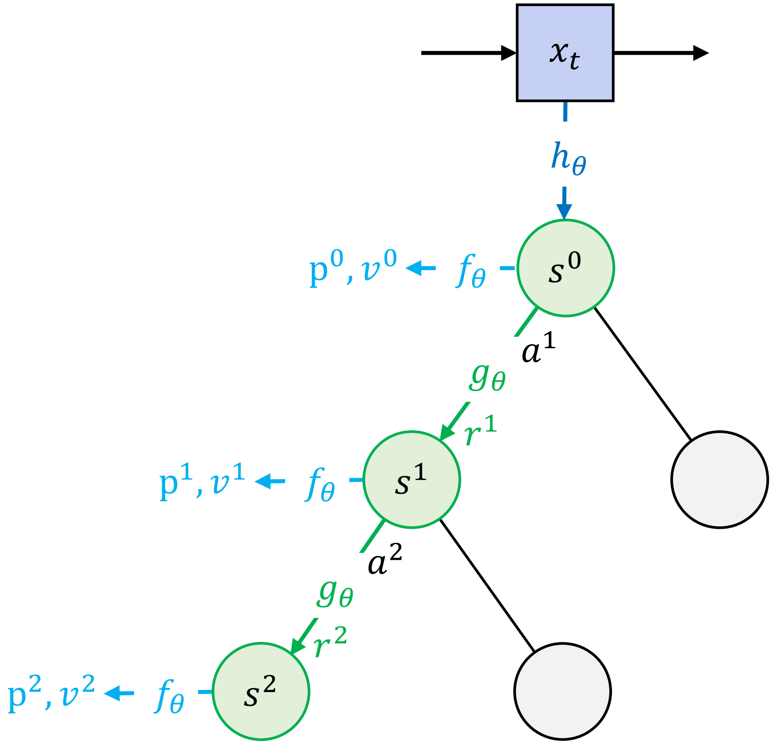


ideal



MuZero (Schrittwieser et al., 2020)

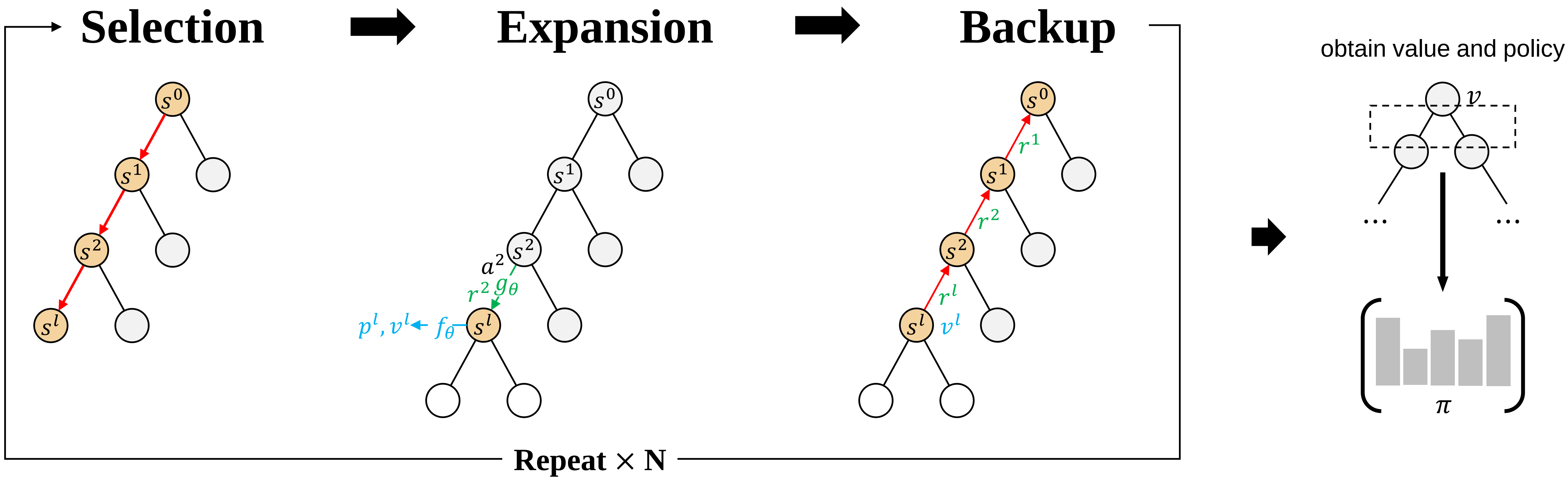
- Achieve superhuman performance on Atari, Go, chess, and shogi
  - Learn from scratch without any human knowledge
  - Plans with a learned model
- Network design
  - Representation: $s^0 = h_\theta(x_1, \ldots, x_t)$
  - Prediction: $p^k, v^k = f_\theta(s^k)$
  - Dynamics: $r^k, s^k = g_\theta(s^{k-1}, a^k)$



x_t: observation
s^k: hidden state
p^k: policy
v^k: value
a^k: action
r^k: reward

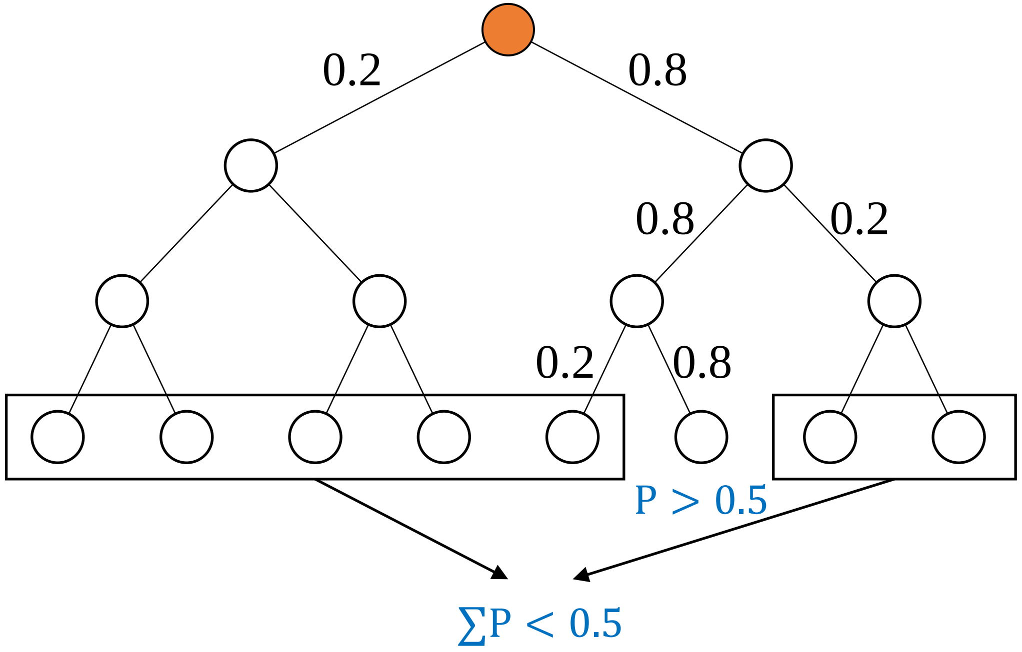
Monte-Carlo Tree Search (MCTS)

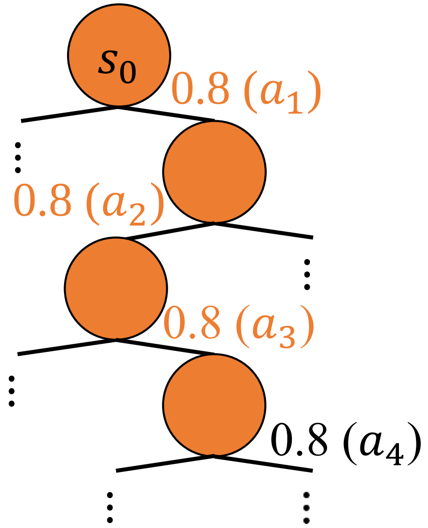
- A heuristic search algorithm
  - In MuZero, it is divided into three phases, repeated for N simulations



Dominant Option

- With max length L, for state s, we define **dominant option** o:
  - $o = \{a_1, a_2, \dots, a_l\}$, where $\prod_{i=1}^{l} P(a_i) > 0.5 \wedge \prod_{i=1}^{l+1} P(a_i) \leq 0.5$ and $l \leq L$
  - The longest action sequence with probability greater than 0.5 ⇒ unique for each state





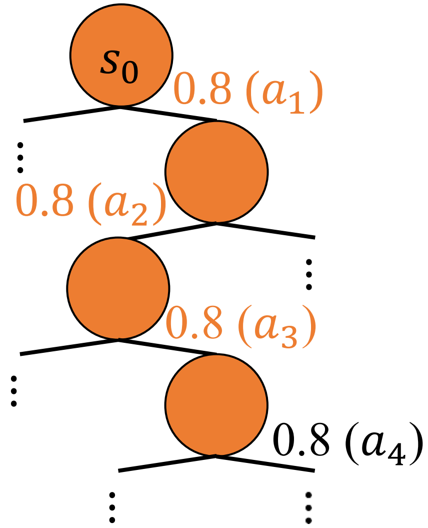
e.g., $L = 4$

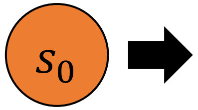
s_0: $o_1 = \{a_1, a_2, a_3\} = \{right, left, right\}$

$P(o_1) = P(a_1)P(a_2)P(a_3) = 0.512$

Option Network

- Given a state and a maximum length L, predict the **dominant option**
  - Output: $\Omega = \{\omega_1, \omega_2, \ldots, \omega_L\}$, each ω_l is a probability distribution, where $\omega_l(a_l) = \prod_{i=1}^{l} P(a_i)$





output of option network

	$left$	$right$	$stop$	
$\boldsymbol{\omega_1}$	0	**0.8**	0.2	$\Rightarrow P(\{right\}) = 0.8$
$\boldsymbol{\omega_2}$	**0.64**	0	0.36	$\Rightarrow P(\{right, left\}) = 0.64$
$\boldsymbol{\omega_3}$	0	**0.512**	0.488	$\Rightarrow P(\{right, left, right\}) = 0.512$
$\boldsymbol{\omega_4}$	0	0.4096	**0.5904**	$\Rightarrow$ terminate

e.g., $L = 4$

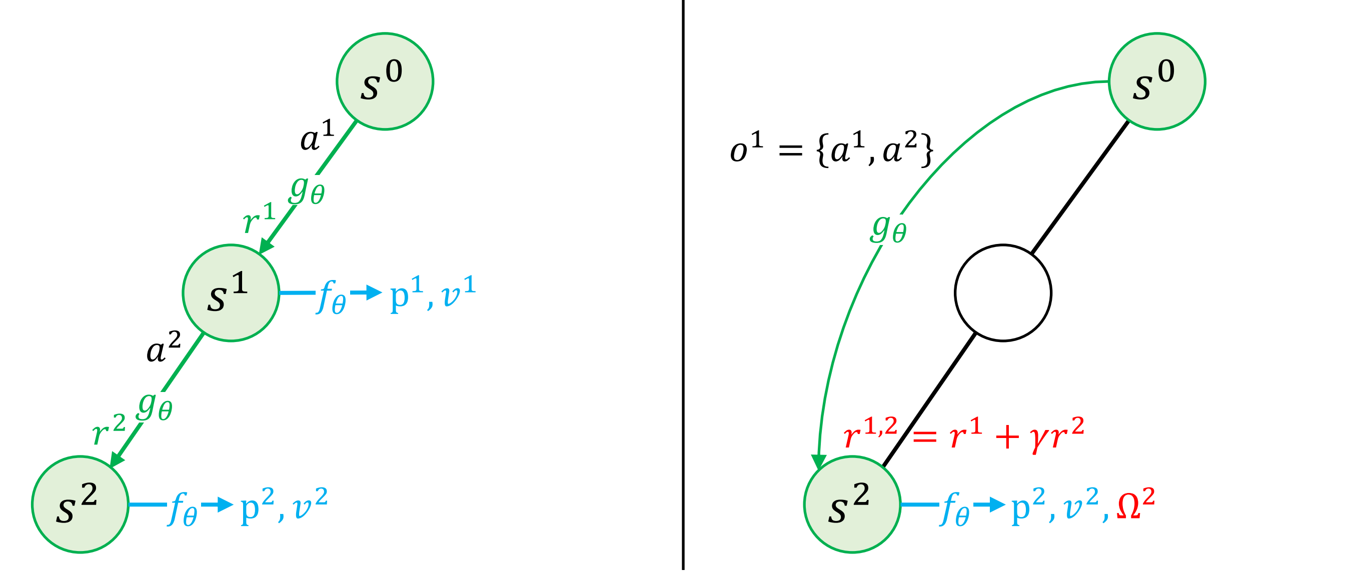
s_0: $o_1 = \{a_1, a_2, a_3\} = \{right, left, right\}$
$P(o_1) = P(a_1)P(a_2)P(a_3) = 0.512$

$o_1 = \{a_1, a_2, a_3\} = \{right, left, right\}$
$P(o_1) = \omega_3(right) = 0.512$

Network Modifications

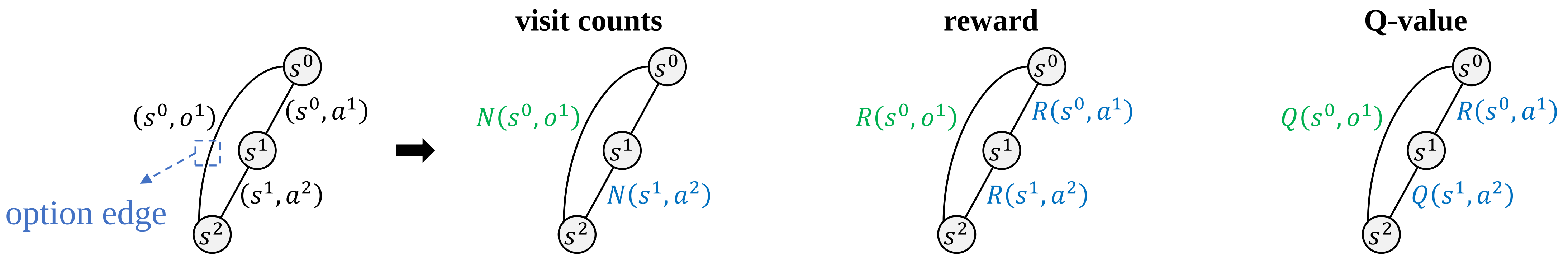
- **Prediction network**: incorporate with **option network**
- **Dynamics network**: predict the outcome of acting an option

MuZero	OptionZero



MCTS Edge Statistics

- Each edge stores a set of statistics: visit counts (N), reward (R), Q-value (Q), prior (P)
- The statistics remain consistent with original MCTS



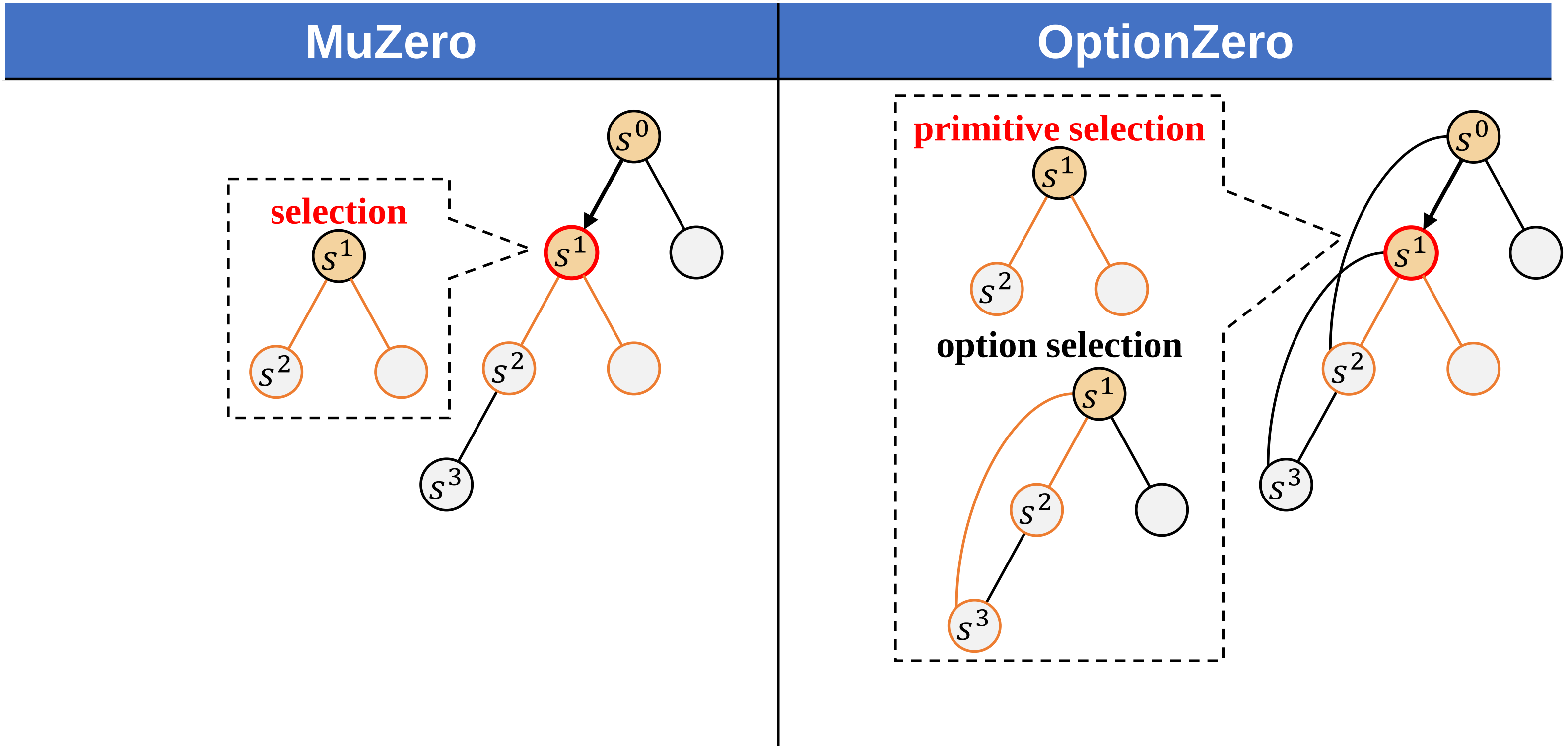
$N(s^0, o^1) = N(s^1, a^2)$

$R(s^0, o^1) = R(s^0, a^1) + \gamma R(s^1, a^2)$

$Q(s^0, o^1) = R(s^0, a^1) + \gamma Q(s^1, a^2)$

Selection – Primitive Selection

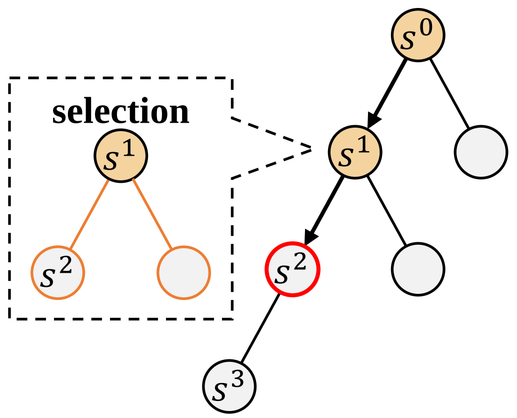
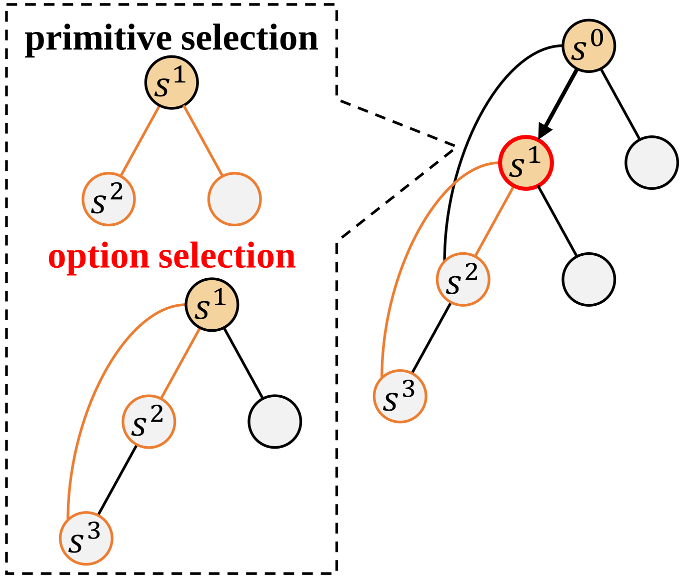
- PUCT score
  - All primitive child nodes: $Q(s^k, a^{k+1}) + P(s^k, a^{k+1}) \times \frac{\sqrt{\Sigma_b N(s^k,b)}}{1+N(s^k,a^{k+1})} \times c_{puct}$



Selection – Option Selection

- PUCT score
  - Option child node: $Q(s^k, o^{k+1}) + P(s^k, o^{k+1}) \times \frac{\sqrt{N(s^k, a^{k+1})}}{1+N(s^k, o^{k+1})} \times c_{puct}$
  - Selected primitive child node: $\tilde{Q}(s^k, a^{k+1}) + \tilde{P}(s^k, a^{k+1}) \times \frac{\sqrt{N(s^k, a^{k+1})}}{1+\tilde{N}(s^k, a^{k+1})} \times c_{puct}$

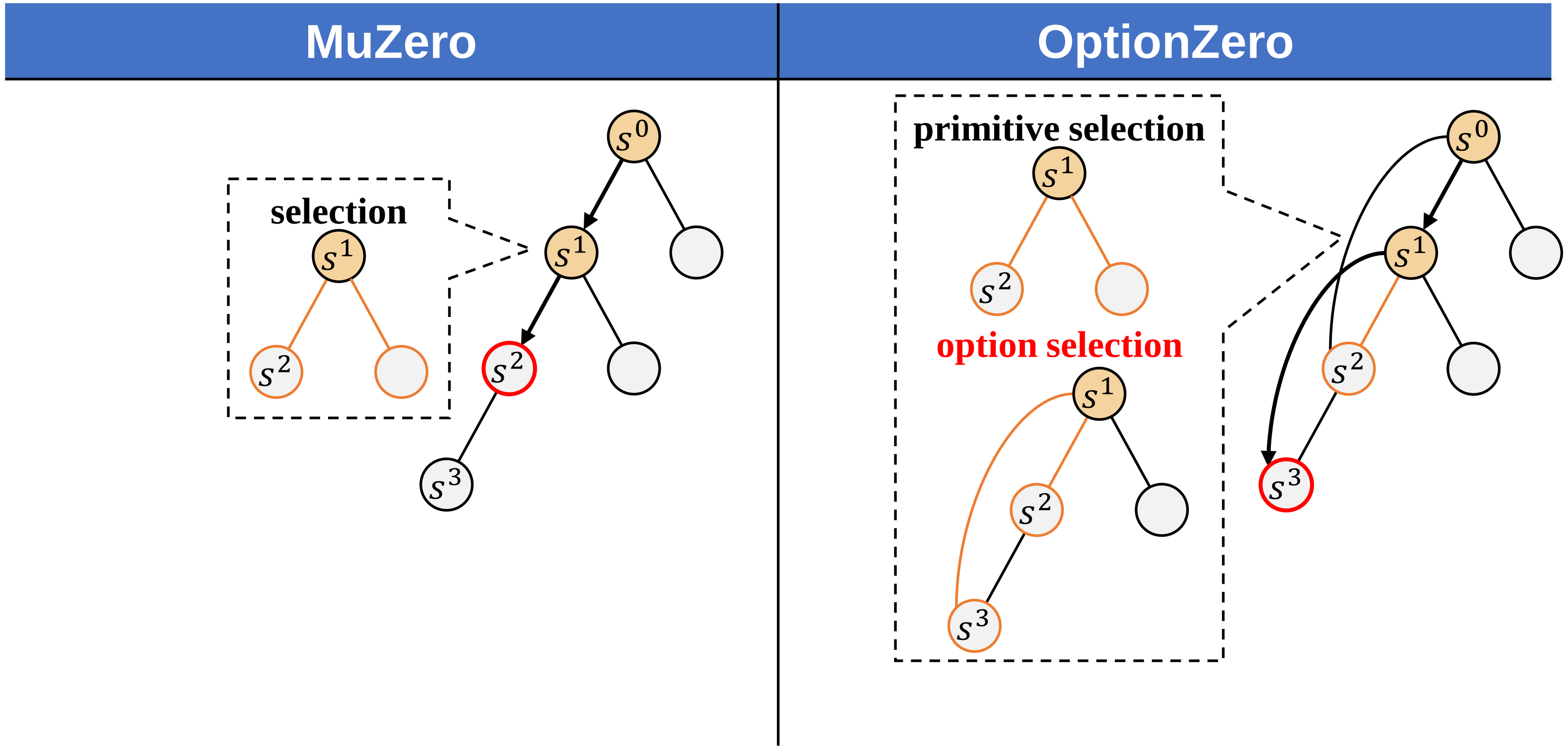
$$\tilde{N}(s^k, a^{k+1}) = N(s^k, a^{k+1}) - N(s^k, o^{k+1})$$
$$\tilde{P}(s^k, a^{k+1}) = \max\left(0, P(s^k, a^{k+1}) - P(s^k, o^{k+1})\right)$$
$$\tilde{Q}(s^k, a^{k+1}) = \frac{N(s^k, a^{k+1})Q(s^k, a^{k+1}) - N(s^k, o^{k+1})Q(s^k, o^{k+1})}{N(s^k, a^{k+1}) - N(s^k, o^{k+1})}$$

MuZero	OptionZero
	

Selection – Option Selection

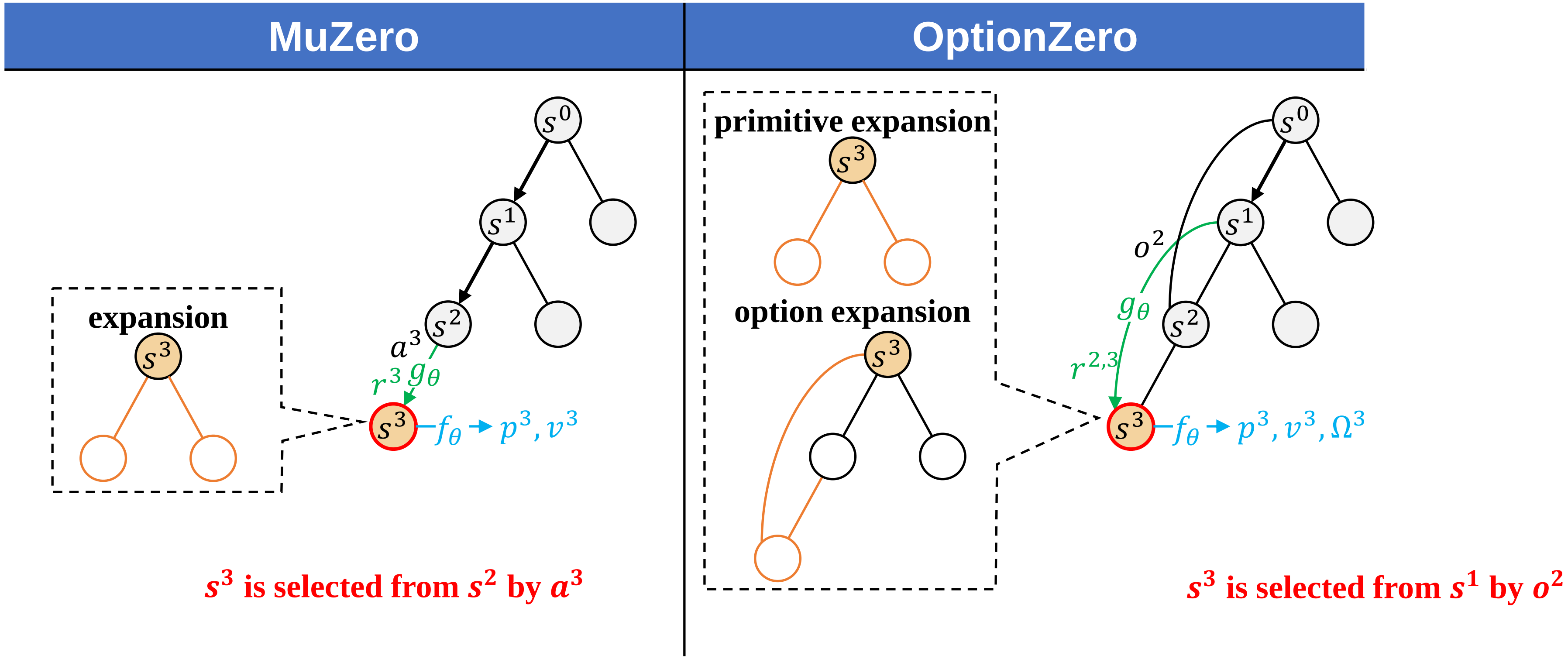
- PUCT score
  - Option child node: $Q(s^k, o^{k+1}) + P(s^k, o^{k+1}) \times \frac{\sqrt{N(s^k, a^{k+1})}}{1+N(s^k, o^{k+1})} \times c_{puct}$
  - Selected primitive child node: $\tilde{Q}(s^k, a^{k+1}) + \tilde{P}(s^k, a^{k+1}) \times \frac{\sqrt{N(s^k, a^{k+1})}}{1+\tilde{N}(s^k, a^{k+1})} \times c_{puct}$

$$\tilde{N}(s^k, a^{k+1}) = N(s^k, a^{k+1}) - N(s^k, o^{k+1})$$
$$\tilde{P}(s^k, a^{k+1}) = \max\left(0, P(s^k, a^{k+1}) - P(s^k, o^{k+1})\right)$$
$$\tilde{Q}(s^k, a^{k+1}) = \frac{N(s^k, a^{k+1})Q(s^k, a^{k+1}) - N(s^k, o^{k+1})Q(s^k, o^{k+1})}{N(s^k, a^{k+1}) - N(s^k, o^{k+1})}$$



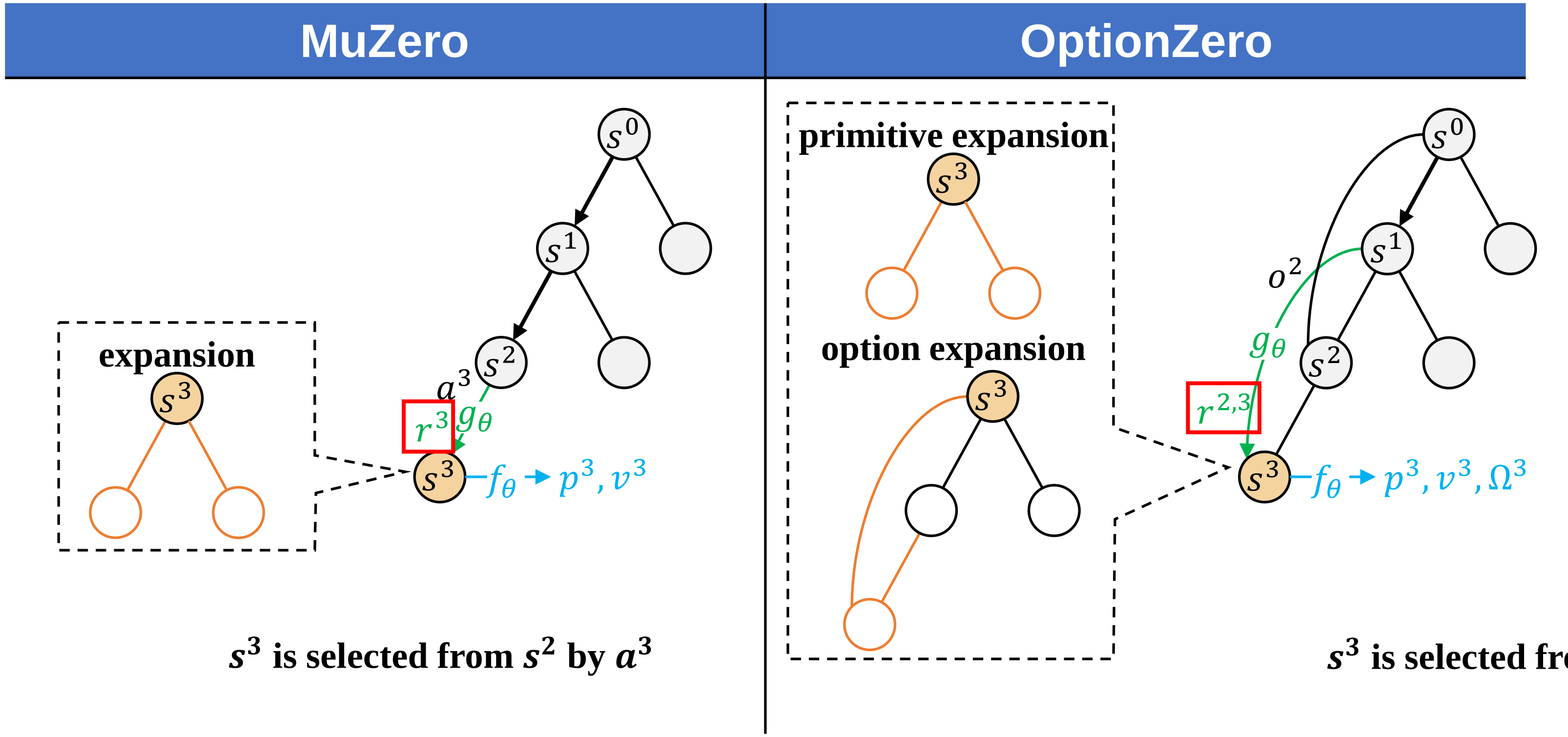
Expansion

- Evaluate the node



Expansion

- Set the reward

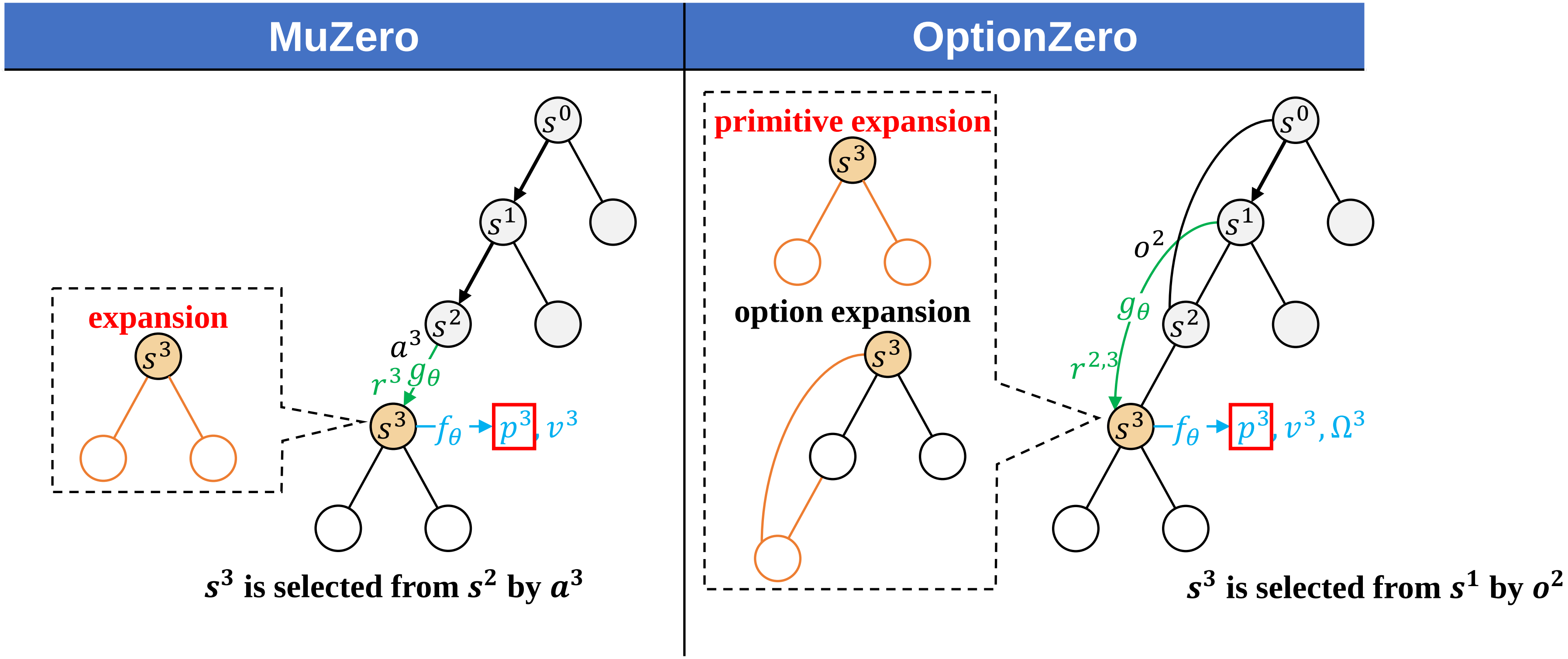


s^3 **is selected from** s^2 **by** a^3

s^3 **is selected from** s^1 **by** o^2

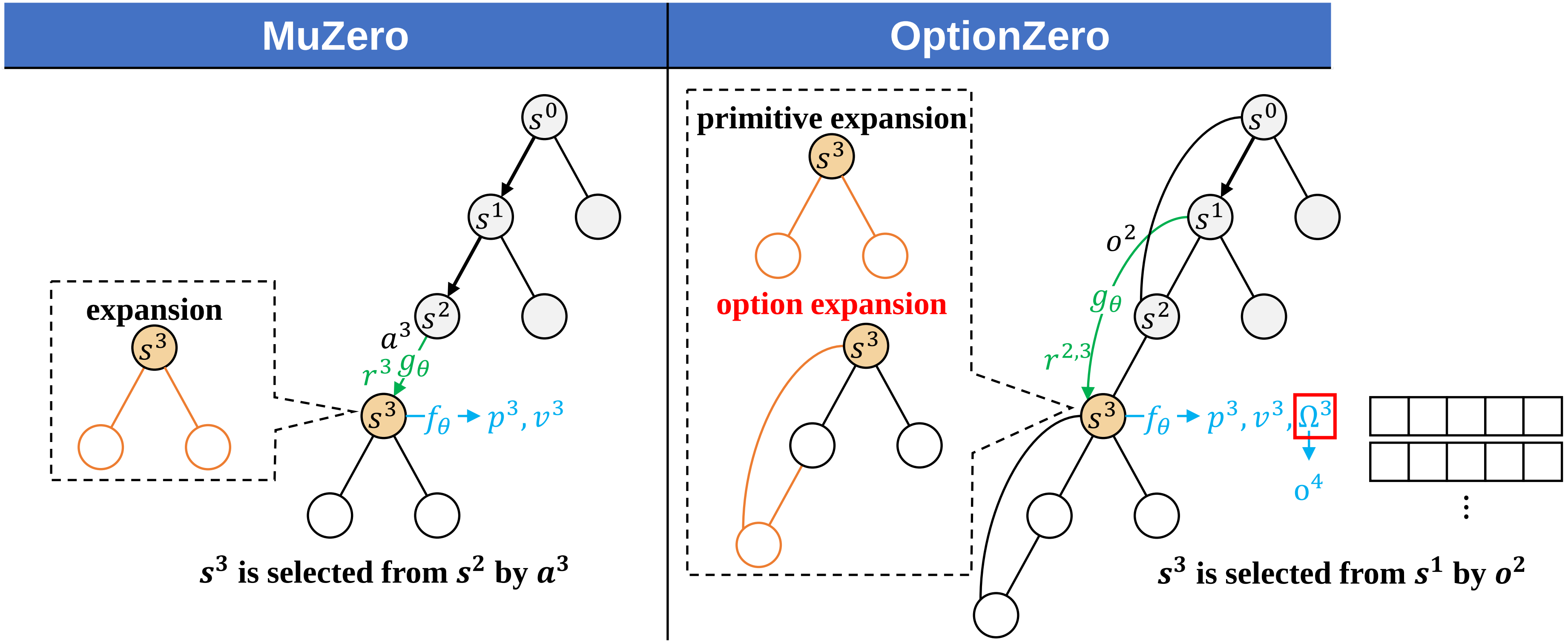
Expansion – Primitive Expansion

- Expand the primitive edges and initialize their prior probabilities



Expansion – Option Expansion

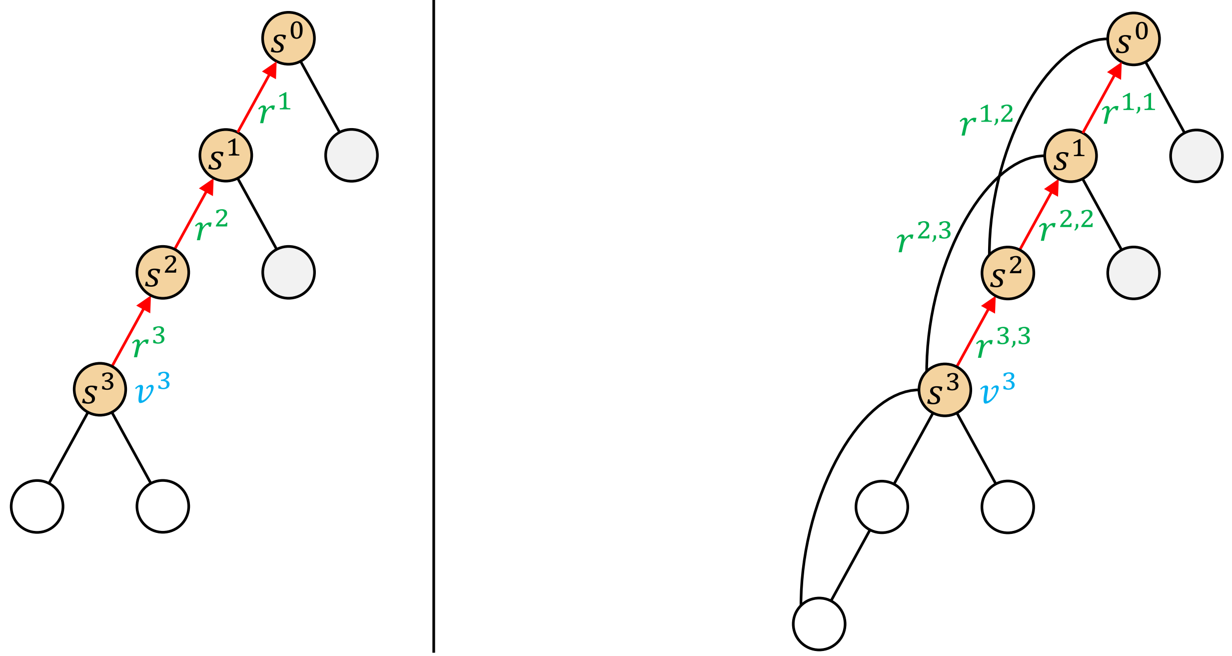
- Expand the option edge and initialize its prior probability



Backup – Update Primitive Edges

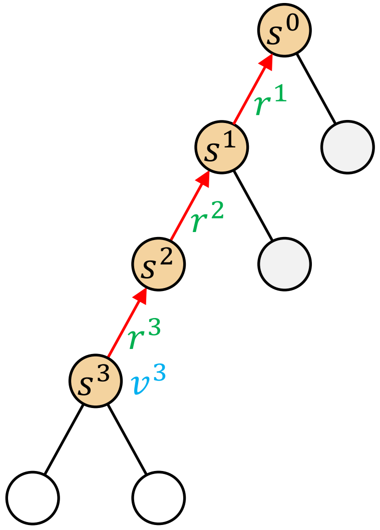
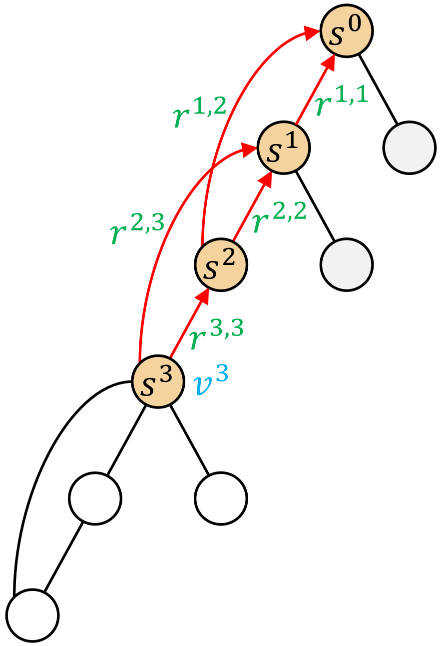
- Update primitive edges on the path from s^0 to s^l
  - $Q(s^k, a^{k+1}) \coloneqq \frac{N(s^k,a^{k+1}) \times Q(s^k,a^{k+1}) + G^{k+1}}{N(s^k,a^{k+1})+1}$, $G^{k+1} = r^{k+1,l} + \gamma^{l-k} v^l$
  - $N(s^k, a^{k+1}) \coloneqq N(s^k, a^{k+1}) + 1$

MuZero	OptionZero

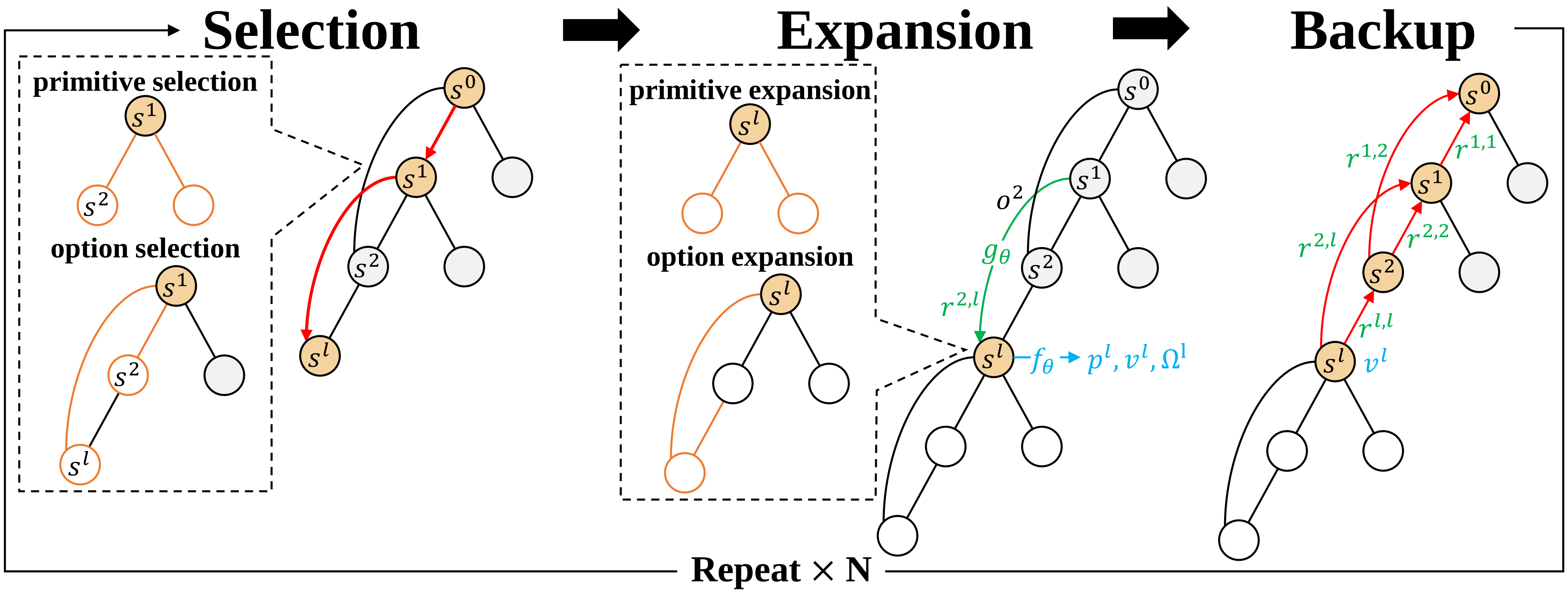


Backup – Update Option Edges

- Update option edges on the possible paths from s^0 to s^l
    - $Q(s^k, o^{k+1}) \coloneqq \frac{N(s^k,o^{k+1}) \times Q(s^k,o^{k+1}) + G^{k+1}}{N(s^k,o^{k+1})+1}$, $G^{k+1} = r^{k+1,l} + \gamma^{l-k} v^l$
    - $N(s^k, o^{k+1}) \coloneqq N(s^k, o^{k+1}) + 1$
    - Ensure the statistics of various selection paths from s^0 to s^l remain consistent

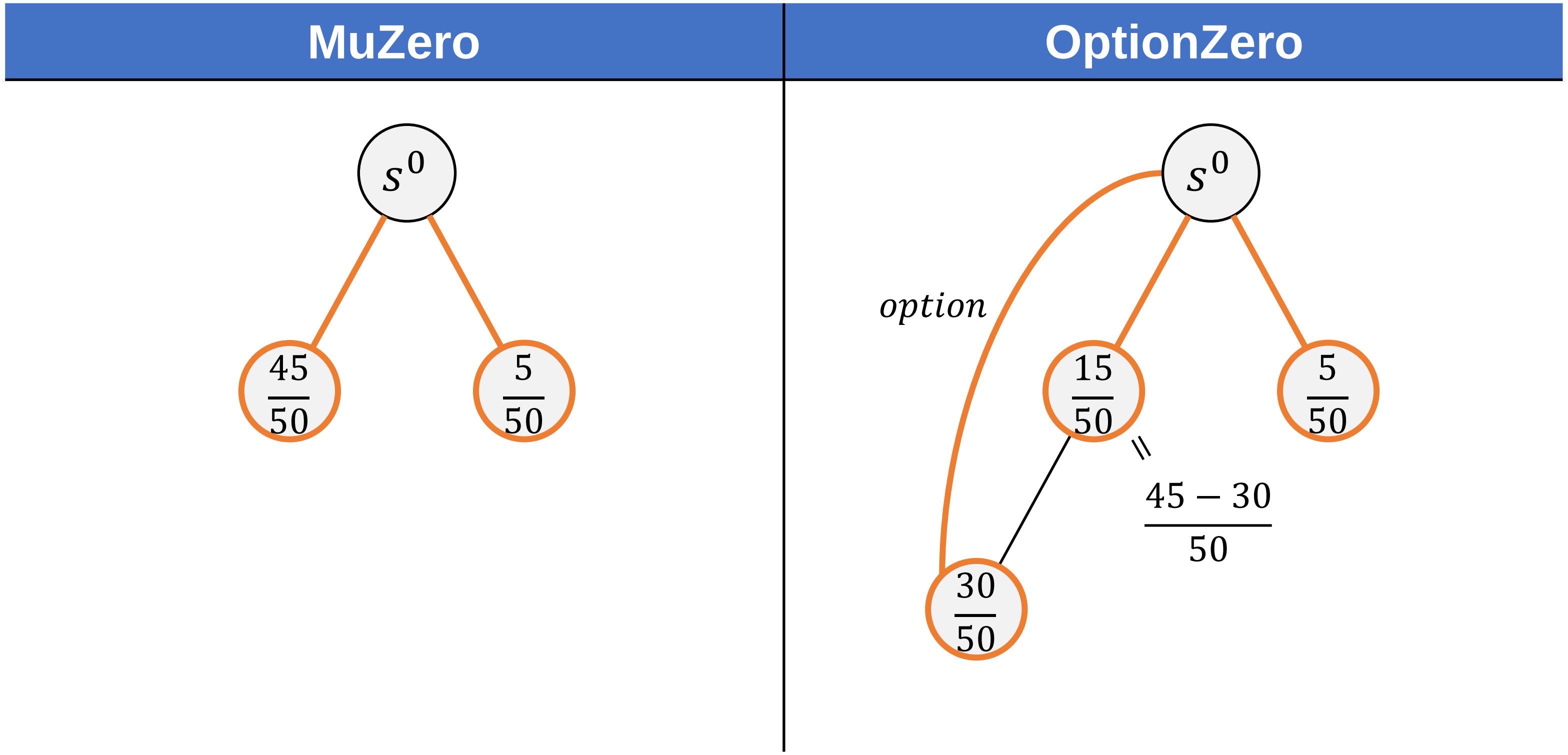
MuZero	OptionZero
	

Planning with Options in MCTS



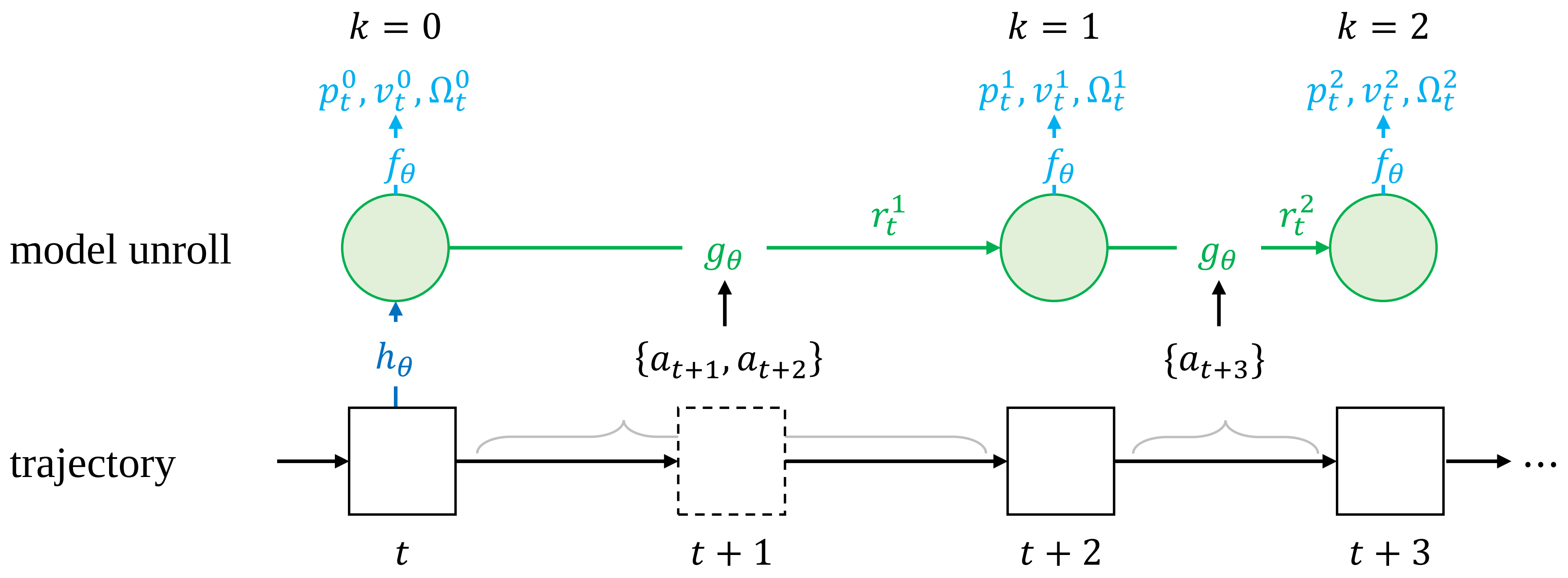
Sample an Action or Option to Execute

- Select a child node of root (s^0) based on probabilities proportional to visit counts
  - Should exclude the duplicate statistics of primitive and option child node



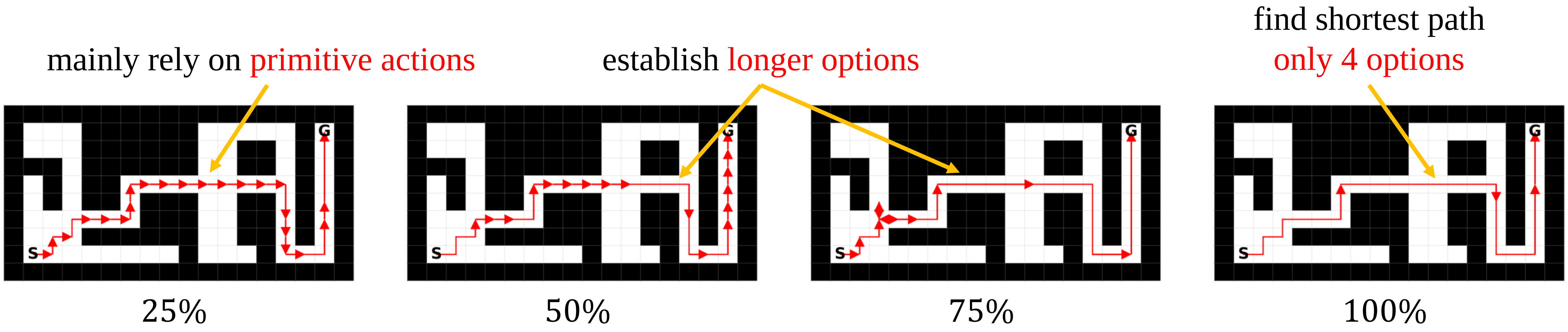
Training

- Optimize the network by self-play trajectories
  - Policy: MCTS policy
  - Value: n-step return or game outcome
  - Reward: observed reward
  - Option policy: actions in the self-play trajectories



GridWorld

- A toy environment to navigate an agent to the goal
- Settings
  - Max option length: $L = 9$
  - Reward of each move: -1



Atari Games

- Environment: 26 Atari games
- Settings
  - ℓ_1: baseline (identical to MuZero), max option length = 1
  - ℓ_3: max option length = 3
  - ℓ_6: max option length = 6
- Results:
  - ℓ_3 performs the best, and both ℓ_3 and ℓ_6 performs better than ℓ_1
    - ℓ_6 performs slightly worse than ℓ_3

      ⇒ the difficulty of learning dynamics network

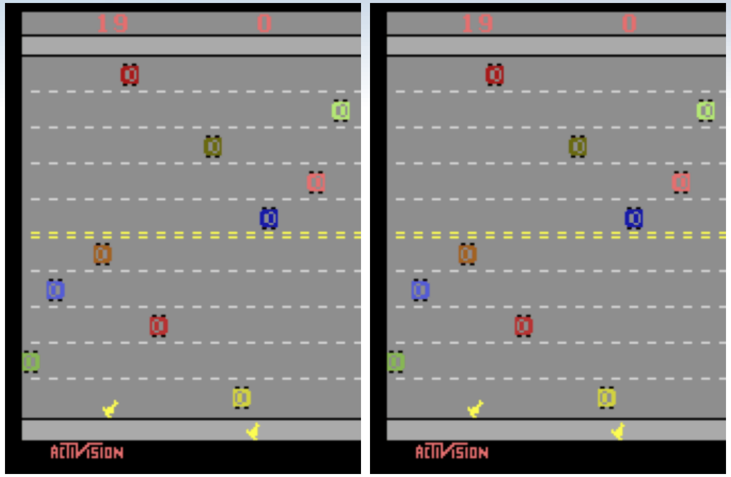
ℓ_i: max option length = i

Bold text in ℓ_3 and ℓ_6 indicates scores that surpass ℓ_1.

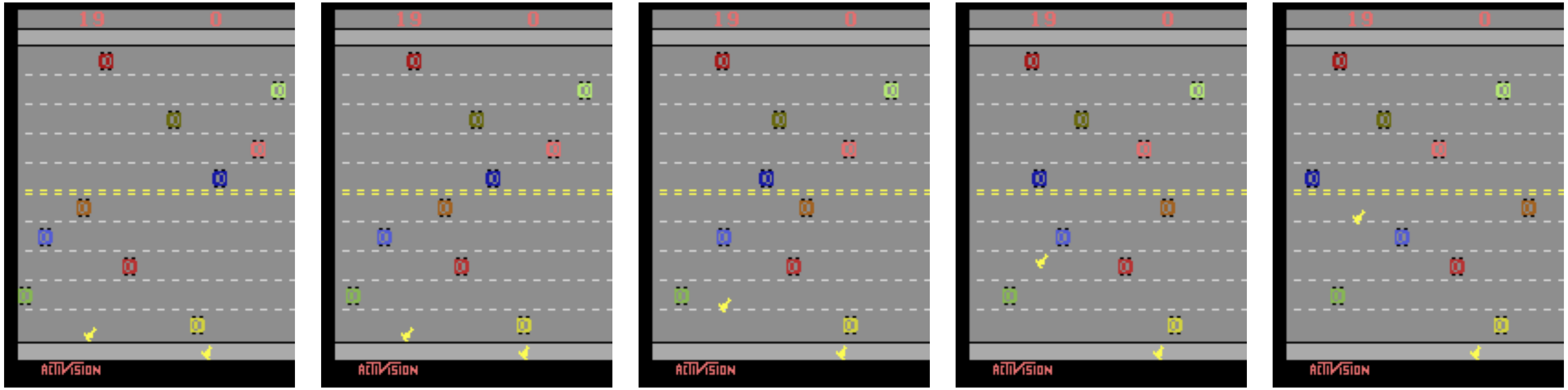
Game	OptionZero ℓ_1	OptionZero ℓ_3	OptionZero ℓ_6
alien	2,437.30	**2,900.07**	**3,748.73**
amidar	780.26	**820.77**	**862.17**
assault	18,389.88	**19,302.04**	**21,593.53**
asterix	177,128.50	**188,999.00**	**187,716.00**
bank heist	1,097.63	950.13	906.53
battle zone	53,326.67	**53,583.33**	39,556.67
boxing	97.71	95.09	96.00
breakout	371.30	**375.58**	364.11
chopper command	43,951.67	**60,181.67**	**45,518.67**
crazy climber	110,634.00	**114,390.00**	**128,455.67**
demon attack	103,823.17	**117,270.57**	**109,092.33**
freeway	29.46	**31.06**	**31.45**
frostbite	3,183.40	**3,641.10**	**6,047.97**
gopher	70,985.27	68,240.60	63,951.47
hero	13,568.20	**19,073.18**	**19,919.22**
jamesbond	8,155.50	**13,276.67**	**8,571.17**
kangaroo	8,929.67	**12,294.00**	**13,951.33**
krull	10,255.37	10,098.83	9,587.57
kung fu master	66,304.67	**68,528.33**	**69,452.33**
ms pacman	3,695.60	**4,952.37**	**4,706.63**
pong	19.37	15.49	17.39
private eye	116.83	90.76	83.24
qbert	17,155.50	**30,748.42**	**36,328.08**
road runner	26,971.33	**32,786.67**	21,699.67
seaquest	3,592.53	**5,606.63**	**6,754.50**
up n down	217,021.60	**280,832.43**	**360,336.33**
Mean (%)	922.72%	**1054.30%**	**1025.56%**
Median (%)	328.40%	**391.69%**	**329.77%**

Behavior Analysis – Repeated Actions

- Game: $freeway$
  - Repeated U: most commonly used options
  - Repeated N: used at certain state



primitive action **option**

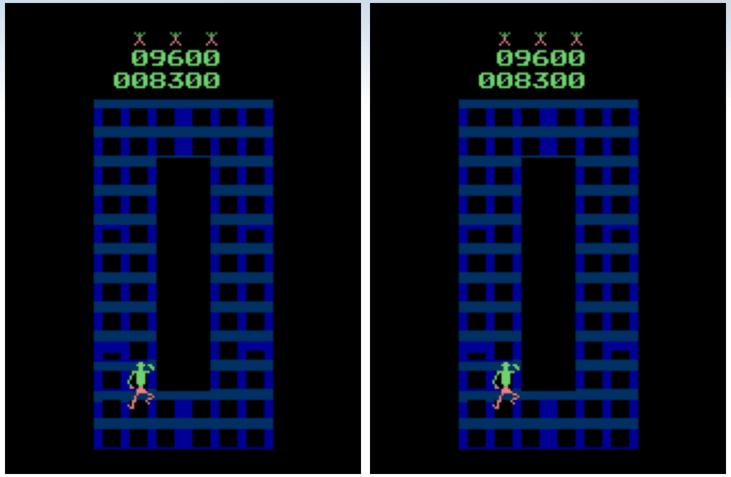


(a) N-N-N-N-N-N (b) D-U-U-U-U-U (c) U-U-U-U-U-U (d) U-U-U-U-U-U (e) D-D-U-U-U-U

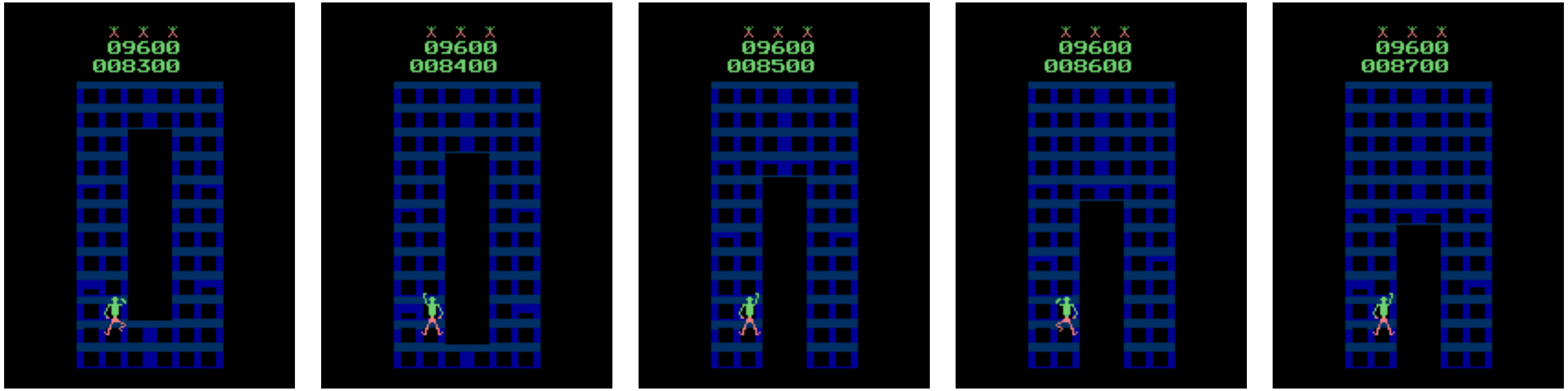
N, U, and D represent no operation, moving up, and moving down, respectively

Behavior Analysis – Non-repeated Actions

- Game: $crazy\ climber$
  - Interleaved U and D: used to climb up



primitive action **option**

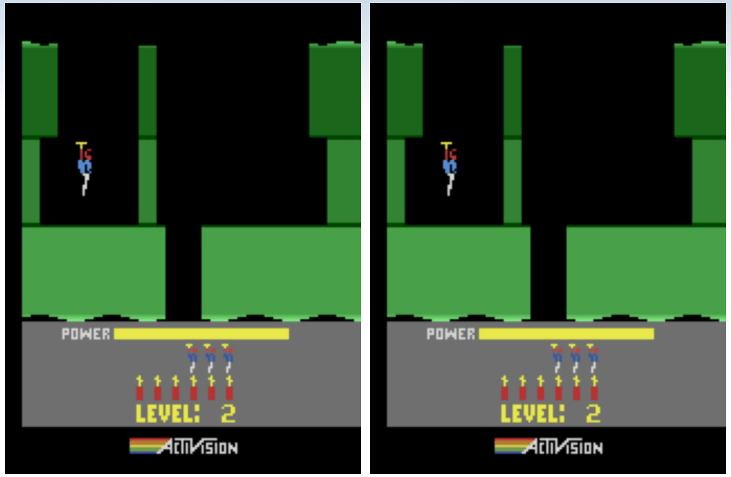


(a) U-U-DL-DL-DL-U (b) U-U-DL-DL-DL-U (c) U-U-DL-DL-DL-U (d) U-U-DL-D-D-U (e) U-U-DL-D-D-U

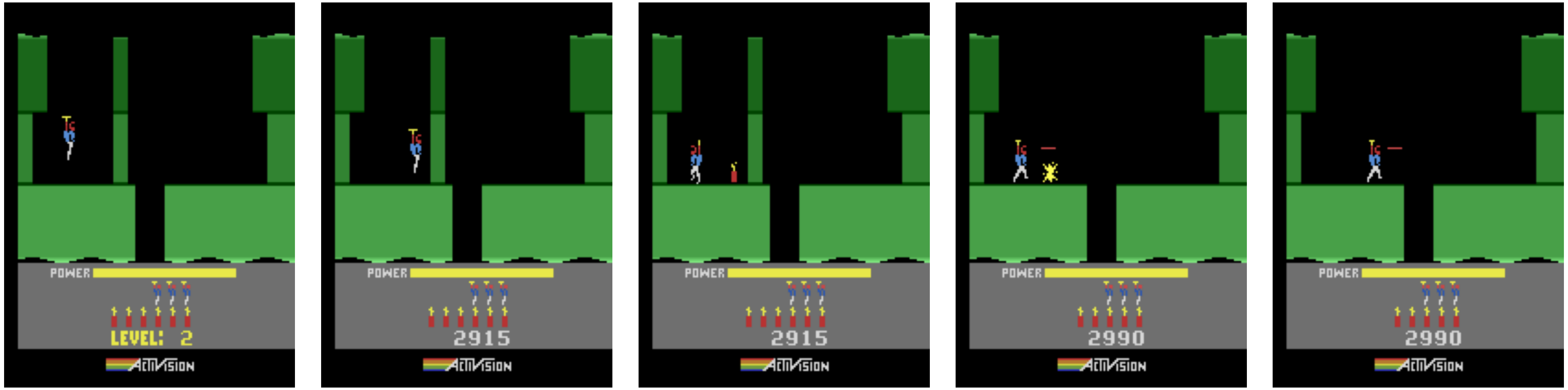
U, D, and L represent control hands, control feet, and moving left, respectively

Behavior Analysis – Strategical Combo

- Game: *hero*
  - Strategical action combinations: break the wall without getting hurt



primitive action **option**



(a) *RF-RF-RF-RF-D-D* (b) *D-L-L-L-L-L* (c) *L-L-RF-RF-RF-RF* (d) *RF-RF-RF-RF-RF-RF* (e) *RF-RF-RF-RF-RF-RF*

R, L, D, and F represent moving right, moving left, placing bombs, and firing, respectively

Option Utilization Analysis

30%~40% using options

Option Utilization in Environment

	% a	% o	% 2	% 3	% 4	% 5	% 6
ℓ_3	62.38%	37.62%	6.23%	31.39%	-	-	-
ℓ_6	69.43%	30.57%	8.55%	3.52%	1.86%	0.99%	15.64%

primitive action/option

% l: the proportion of option length l

Option Utilization in the Search

	Avg. tree depth	Median tree depth	Max tree depth
ℓ_1	14.52	12.58	48.54
ℓ_3	20.74	18.23	121.46
ℓ_6	24.92	19.35	197.58

option allows search deeper under same budget

Summary

- OptionZero
  - Integrate options into the MuZero algorithm
  - **Autonomously discover options** through self-play games
  - **Utilize options during planning**
  - Outperform a MuZero baseline on 26 Atari games
- Future work
  - Apply to two-player games
  - Integrate with other dynamics models

Thank You for Your Attention

Our code and data are available at

https://rlg.iis.sinica.edu.tw/papers/optionzero/