NEURAL INFORMATION PROCESSING SYSTEMS

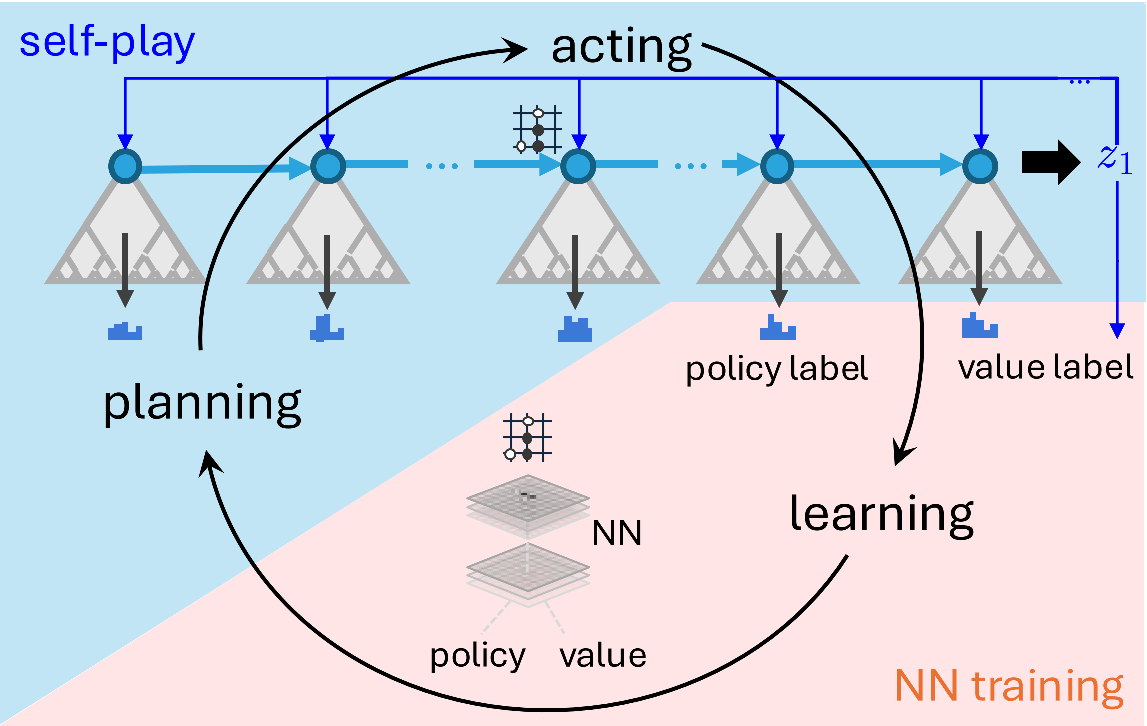
Uncertainty-Guided Exploration for Efficient AlphaZero Training

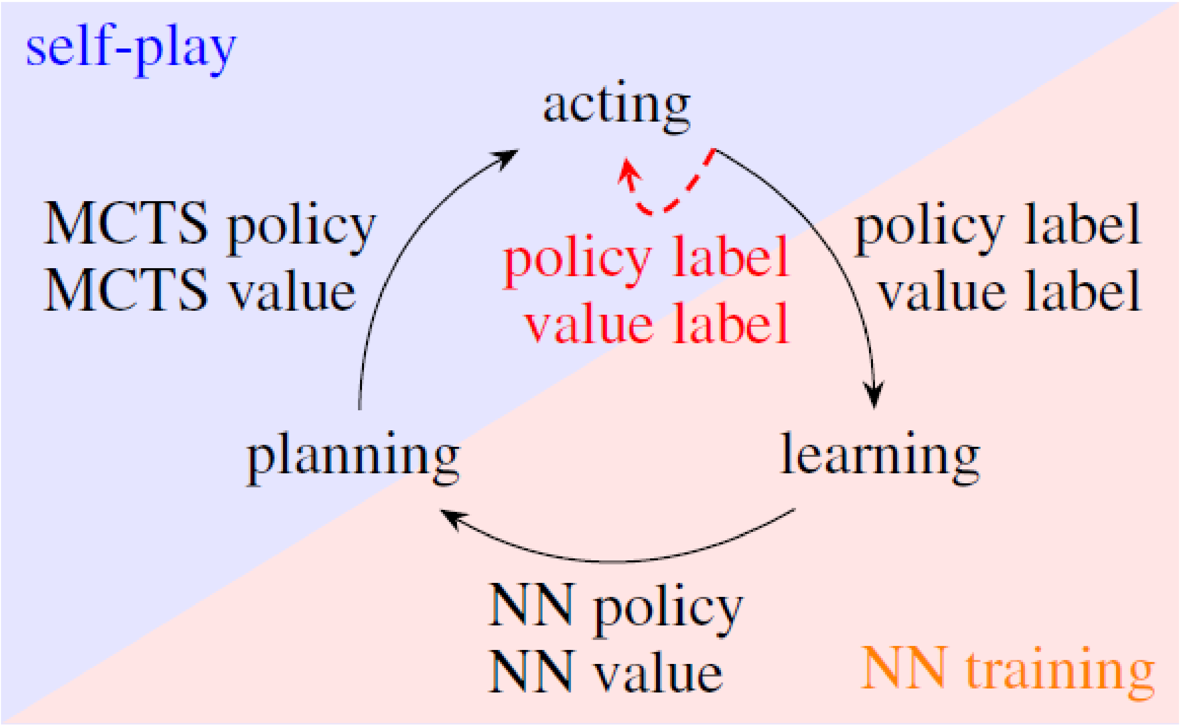
Scott Cheng[1], Meng-Yu Tsai[2], Ding-Yong Hong[3], Mahmut Taylan Kandemir[1]

[1] The Pennsylvania State University, USA
[2] Independent
[3] Academia Sinica, Taiwan

Background: AlphaZero

Many recent breakthroughs in artificial intelligence have been driven by the AlphaZero algorithm, which consists of self-play and NN training:

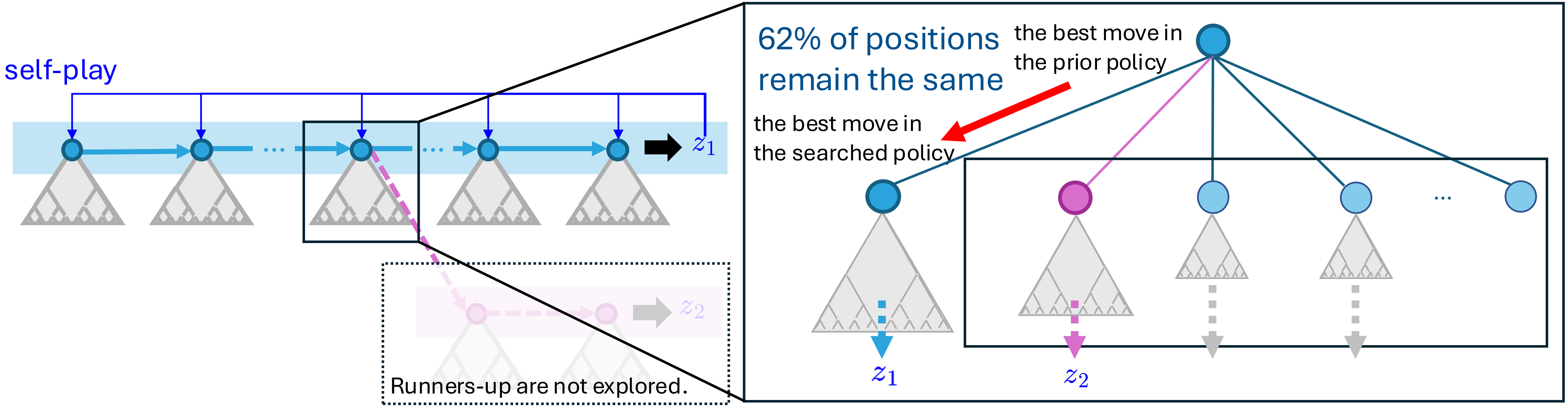




AlphaZero's self-play process remains inefficient

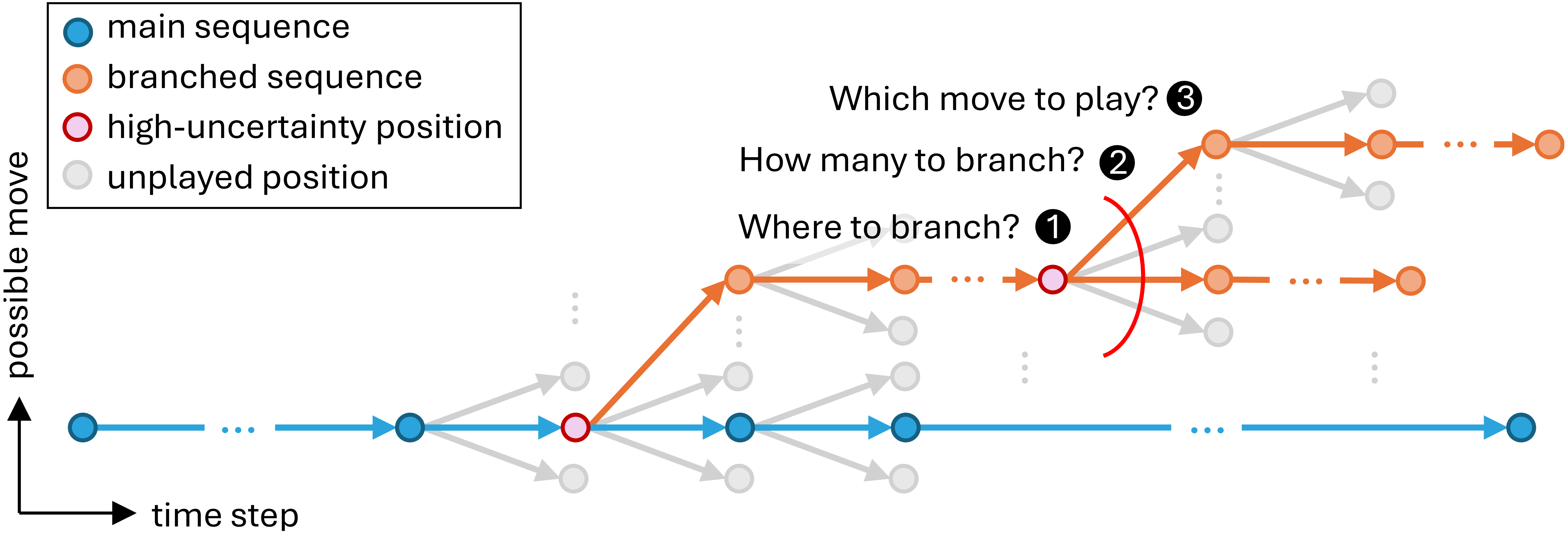
(1) Value labels have high variance, as they are derived from a single result.

(2) A few positions exhibit high-uncertainty, but these positions are not further explored.



(3) When deeper search changes the best move, 79% of the new best moves come from the top-3 runners-up.

Uncertainty-Guided Exploration during Self-play



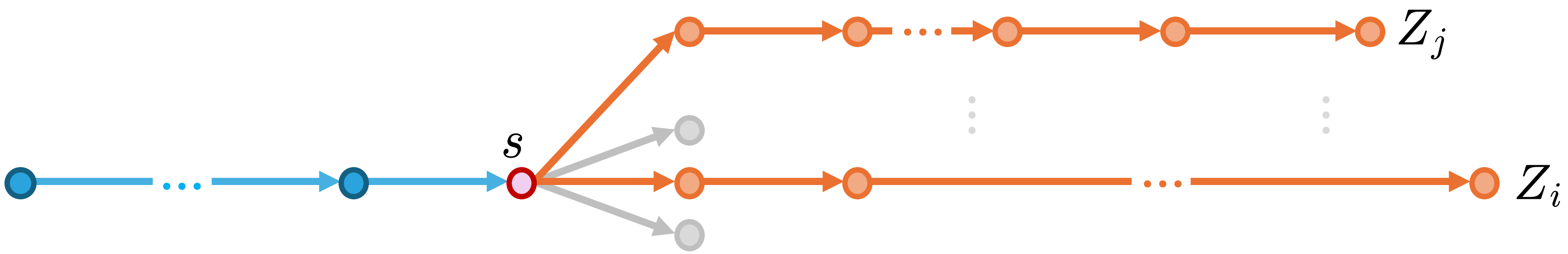
Label Change Rate (LCR)

Game result: $Z \sim \mathrm{Bernoulli}(w)$ Game plays: $Z_1, \ldots, Z_n \overset{\text{i.i.d.}}{\sim} Z$

Uncertainty metric: $\theta = \mathrm{LCR}(s)$

For any independent plays $i \neq j$,

$$\mathrm{LCR}(s) := \Pr(Z_i \neq Z_j) = 2w(1-w)$$



$\hat{\theta} = 2v(1-v)$ for MCTS value $v \in [0, 1]$

Bayesian Inference

Bayesian view: model the LCR θ as a random variable.

Label change: $X = \mathbf{1}_{Z_i \neq Z_j}$

Observed data: $D = \{X_1, X_2, \dots, X_m\}$ where $X_i \sim \mathrm{Bernoulli}(\theta)$

Prior LCR:

$$\mathbb{E}[\theta] = \frac{\alpha}{\alpha + \beta}$$

where $\theta \sim \mathrm{Beta}(\alpha, \beta)$

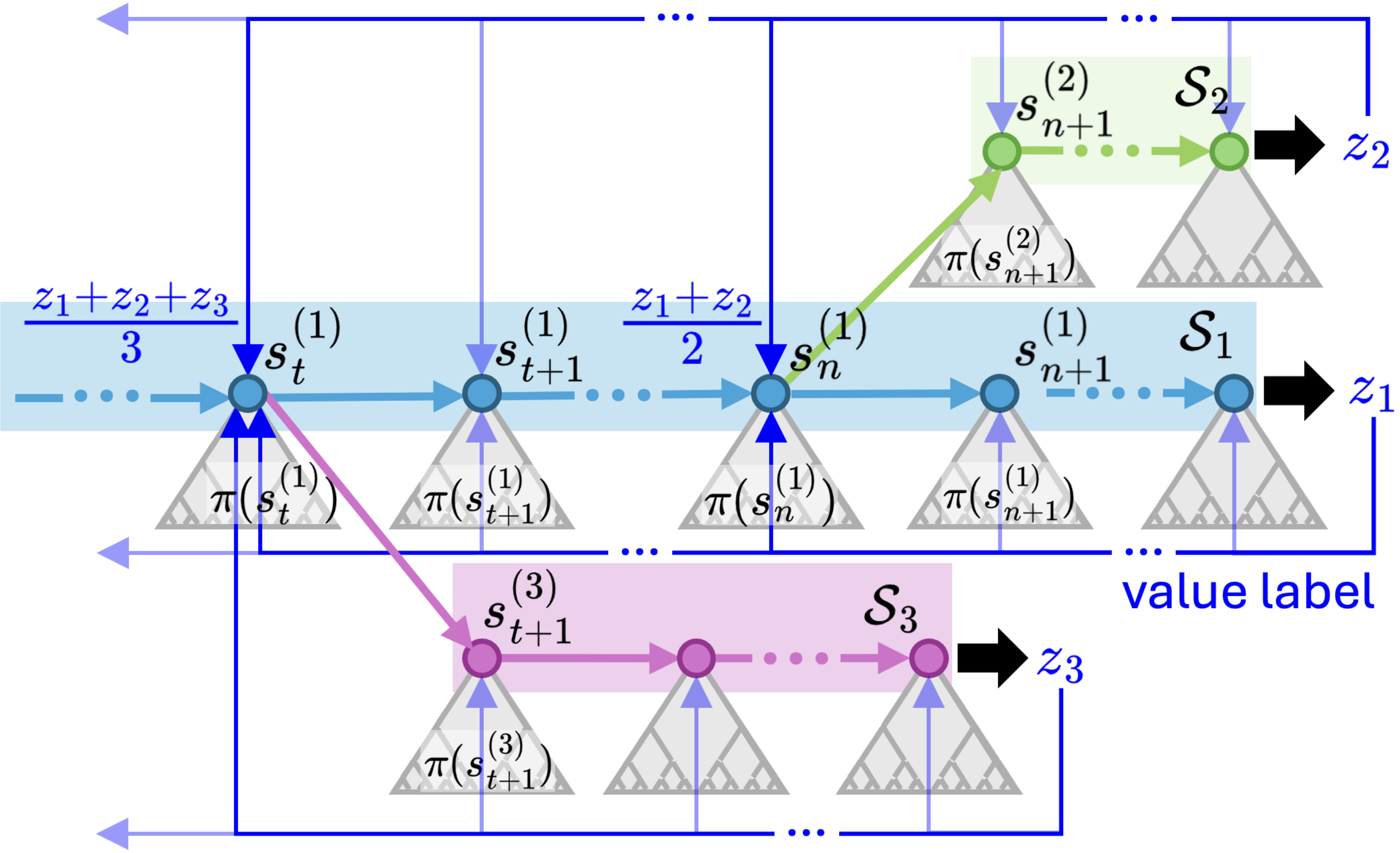
Posterior LCR:

$$\mathbb{E}[\theta \mid D] = \frac{\alpha + s}{\alpha + \beta + m}$$

where $\theta \mid D \sim \mathrm{Beta}(\alpha + s, \beta + m - s)$

with s observed result changes

Uncertainty-Guided Branching



$$\mathrm{Var}[\bar{Z}] = w(1-w)/n$$

Algorithm 1: Uncertainty-Guided Branching

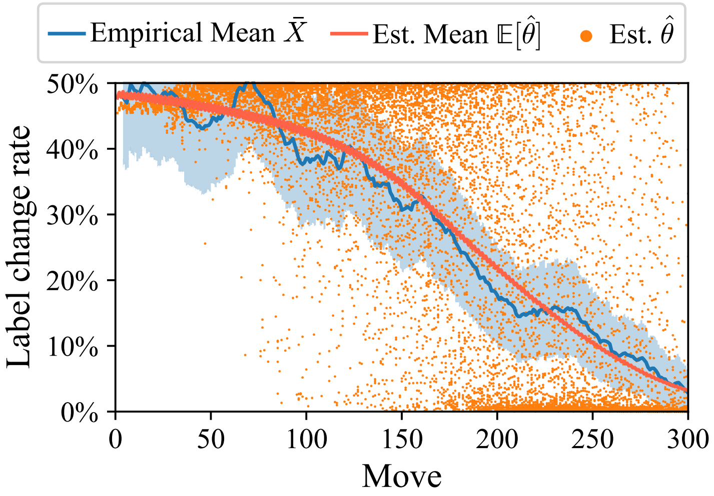
Input: G number of states to collect
Output: $\mathcal{G}$ ={(state, policy label, value label)}

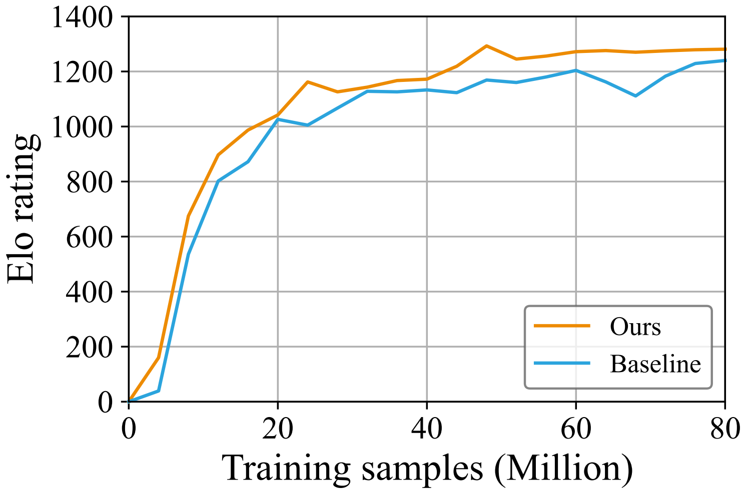
```
G ← ∅
while |G| < G do
    (s, a) ← initial state and action
    for i ← 1 to V do
        {(s′, π(s′), z_i) | s′ ∈ S_i} ←
            play episode from (s, a)
        Compute LCR and sampling weights
         u (defined in Equation 4)
        s ∼ Y(u) (defined in Equation 5)
        a ∼ π( · | s)
        while (s, a) has been played do
            s ← preceding state of s
            a ∼ π( · | s)
    S := ⋃_{i=1}^{V} S_i
    G ← G ∪ {(s′, π(s′), Σ_{i: s′∈S_i} z_i / |{i: s′∈S_i}|) | s′ ∈ S}
return G
```

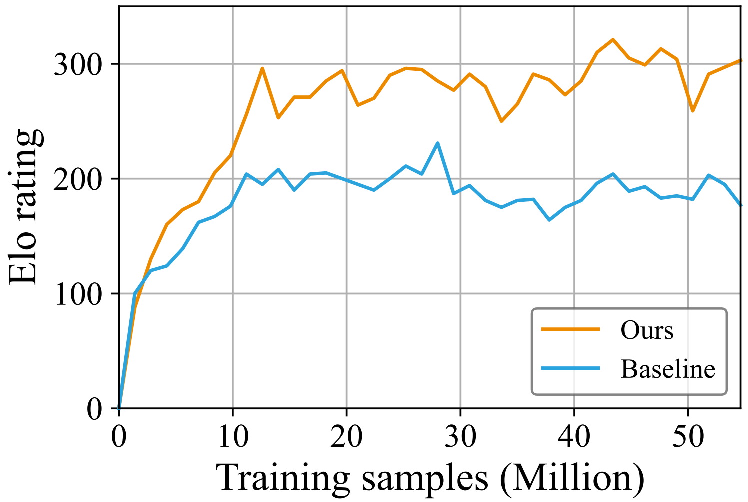
Evaluation

1. End-to-end training: improves sample efficiency up to 58.5% in 9x9 Go and 47.3% in 19x19 Go.
2. LCR uncertainty estimator closely matches the empirical results with an average RMSE of 0.02.
3. Our design space exploration shows that V=10 achieves the best balance for exploration within and between games.

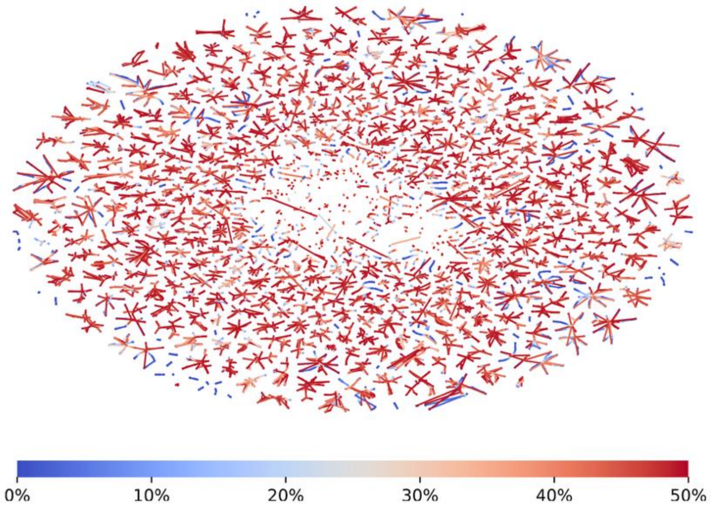
and more...

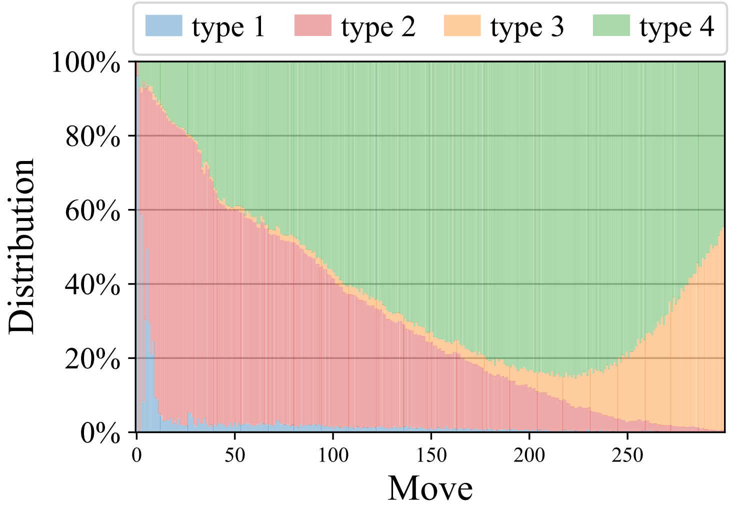


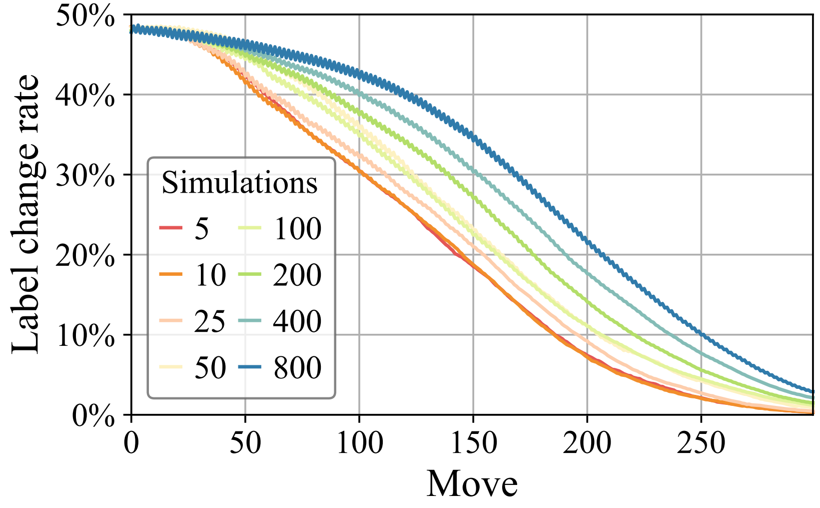


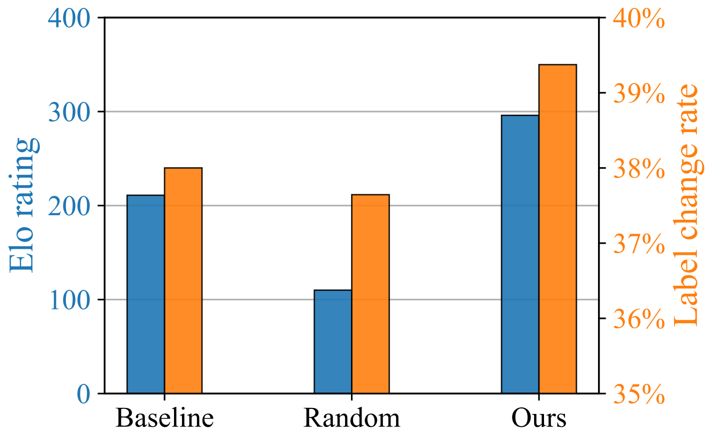


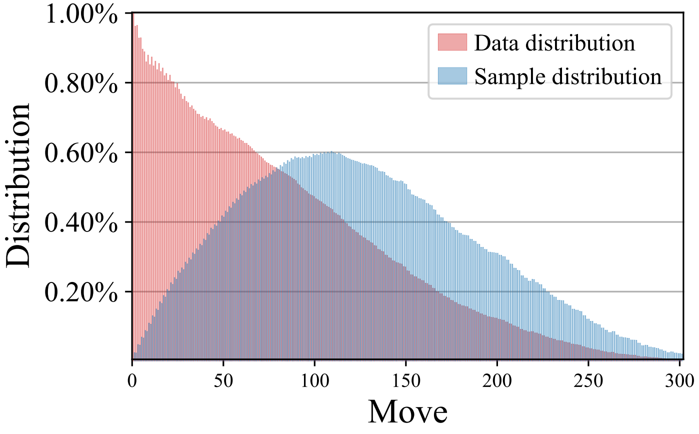
For more details... Please check our paper!

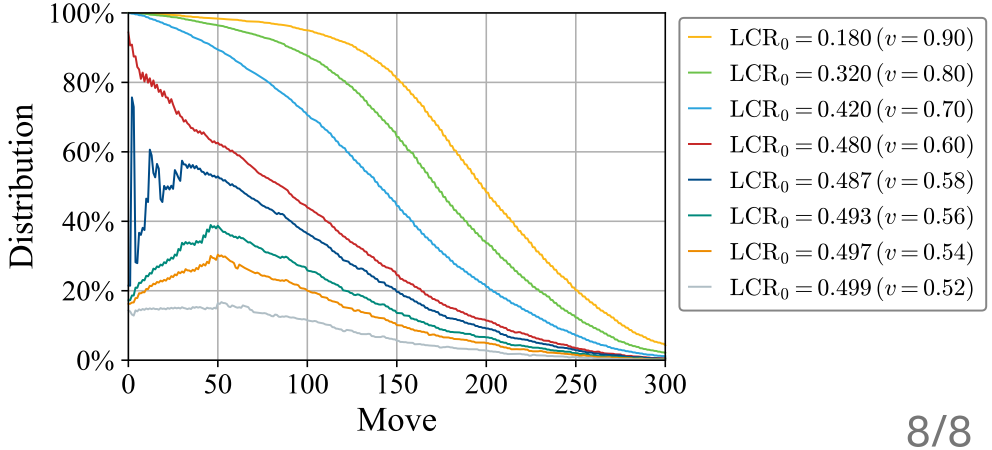












Thank you

Acknowledgements to:

- DOE grant and NSF grants*
- National Science and Technology Council of Taiwan
- Taiwan National Center for High-performance Computing (NCHC)
- Anonymous reviewers for their valuable comments

* The opinions expressed in this paper are those of the authors and not necessarily the views of the NSF or the DOE.