Action Sampling Strategies in *Sampled MuZero* for Continuous Control Tasks

Background

Model-based Reinforcement Learning

- Learns environment rules alongside policies and value functions
- Builds internal models to enable planning and improve sample efficiency
- Example: *Google DeepMind's MuZero [3]* achieved state-of-the-art performance on Atari
- Uses Monte Carlo Tree Search (MCTS) to simulate future trajectories and improve value estimates

Continuous Control Problems

- Actions selected from real-valued, high-dimensional spaces.
- Common in robotics: applying torque to every motorized joint.
- We can factorize the policy to handle continuous actions:

$$\pi(a|s) = \prod_{i=1}^{n} \pi_i(a_i|s) = \prod_{i=1}^{n} \mathcal{N}_i(a_i; \mu_\theta(s), \sigma_\theta(s))$$

- Base MuZero **cannot** handle continuous spaces – Infeasible to represent infinite possible actions as separate nodes in the search tree

Sampled MuZero [2] Modifications

- **MCTS Node Expansion.** Instead of considering all $N = |\mathcal{A}|$ actions, we sample a fixed $K \ll N$ actions from a proposal distribution β
- **PUCT Formula.** To obtain an unbiased estimate of the improved policy the search must use adjusted prior P

$$\arg\max_a Q(s,a) + c \cdot \underbrace{\frac{\hat{\beta}(a,s)\pi(a,s)}{\beta(a,s)}}_{\text{Prior } P} \cdot \frac{\sqrt{\sum_b N(s,b)}}{1 + N(s,a)}$$

Action Sampling Strategies

Sampled MuZero proposes a general framework but leaves room for exploring (1) how actions should be sampled (i.e. from what distribution) and (2) how many actions should be sampled at each node during MCTS

Alternatives for Sampling Distribution β

- **Uniform distribution** i.e. $\beta = U(-1, 1)$.
  - We **sample** actions uniformly
  - We **search** with prior $P = \pi$ (as base MuZero)
- **Temperature modulated policy distribution** $\beta = \pi^{1/\tau}$.
  - We **sample** actions from agent's policy (temperature modulated) $\pi^{1/\tau}$
  - We **search** with policy prior $P = \hat{\beta}\pi^{1-1/\tau}$.
- $\tau = 1$ – We search with a uniform prior $P = \hat{\beta}$ (used in Sampled MuZero)
- $\tau > 1$ – We explore a more diverse set of actions (we add sampling noise), but search is guided by more peaked probabilities. For $\tau < 1$ the opposite is true.
- How temperature τ affects the Gaussian factorized policy we are using:

$$\pi_\theta^{1/\tau} = \prod_i \mathcal{N}_i(\mu_\theta, \sqrt{\tau} \cdot \sigma_\theta)$$

Progressive Widening [1]

- MCTS augmentation that adjusts number of child nodes considered based on parent node visits
- When Instead of considering all K actions at once, we start with C actions and sample additional actions only if `num_children[s]` $< C \cdot$ `num_visits[s]`$^\alpha$
  - C: The base number of nodes/actions we start with.
  - α: Controls how often we sample more actions and widen the tree. Higher α means it takes fewer visits to trigger sampling of an additional action from β.

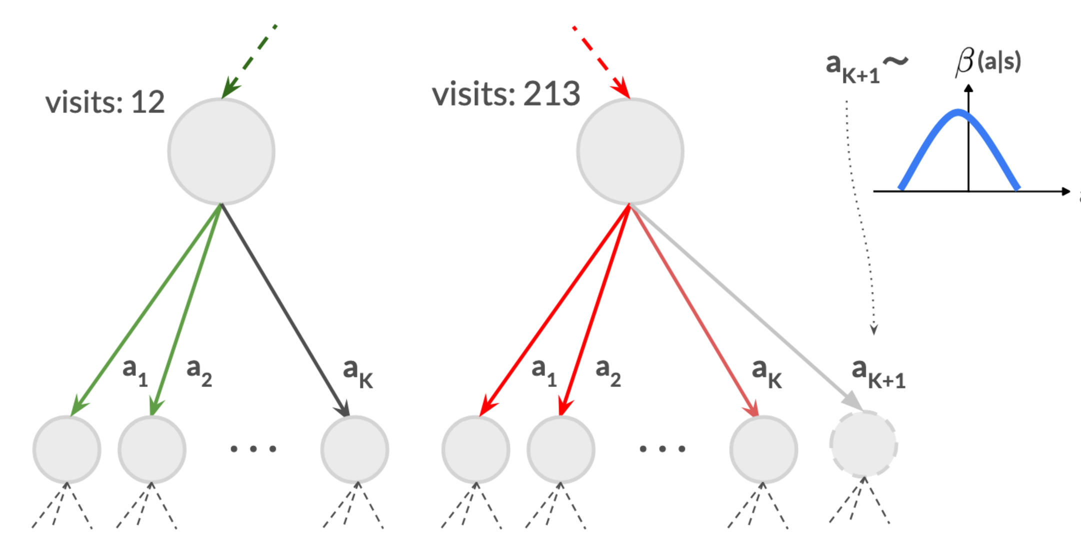

Figure 1. Illustration of progressive widening in MCTS. As the number of visits to a node increases, additional actions are sampled from a continuous distribution $\beta(a|s)$ and added to the node's children, expanding the search space adaptively.

Experimental Setup

- **Physics Engine**: Brax (JAX-based) to complement agent implementation.
- **Environment**: 2D bipedal robot, 17D observations and 6D actions
- **Objective**: Maximize forward speed while minimizing energy consumption
- **Training Duration**: 1M steps (∼15 hours) due to computational constraints
- **Network**: ResNet v2 style with 4 blocks, 512 hidden dimensions, leaky ReLU + layer normalization
- **Policy**: Factored Gaussian with tanh to squash actions into bounds [-1, 1]
- **Action Sampling**: $K = 10$ sampled actions, 50 MCTS simulations per move

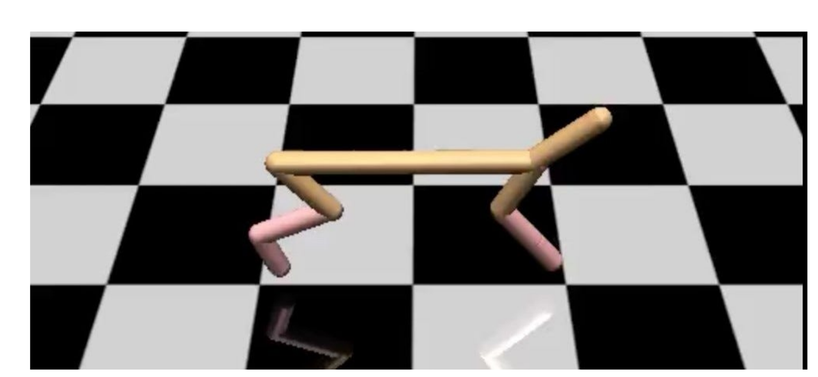

Figure 2. HalfCheetah training envrionment.

Results

Comparison of Proposal Distributions β

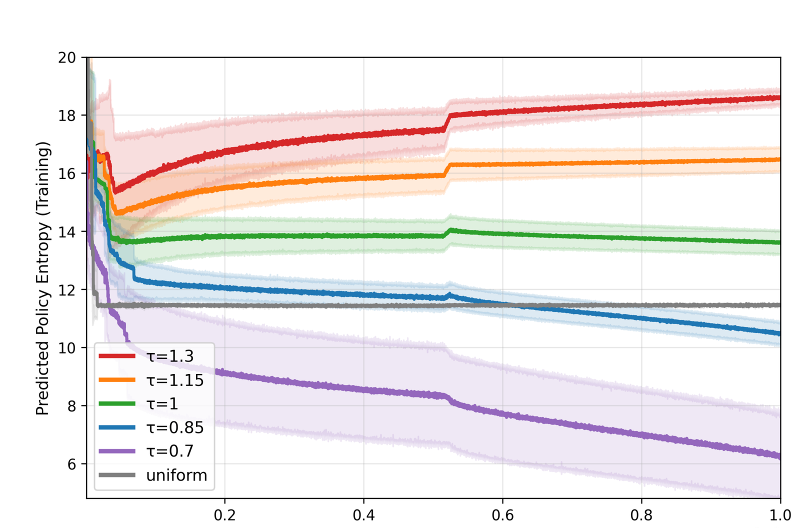

Figure 3. **Policy network entropy across β distributions.** Higher temperatures ($\tau > 1$) maintain elevated entropy throughout training, promoting exploration of diverse actions. Lower temperatures ($\tau < 1$) lead to rapid entropy decay, indicating faster convergence to peaked action distributions. Uniform sampling maintains constant entropy. Shaded regions show standard error over 5 seeds.

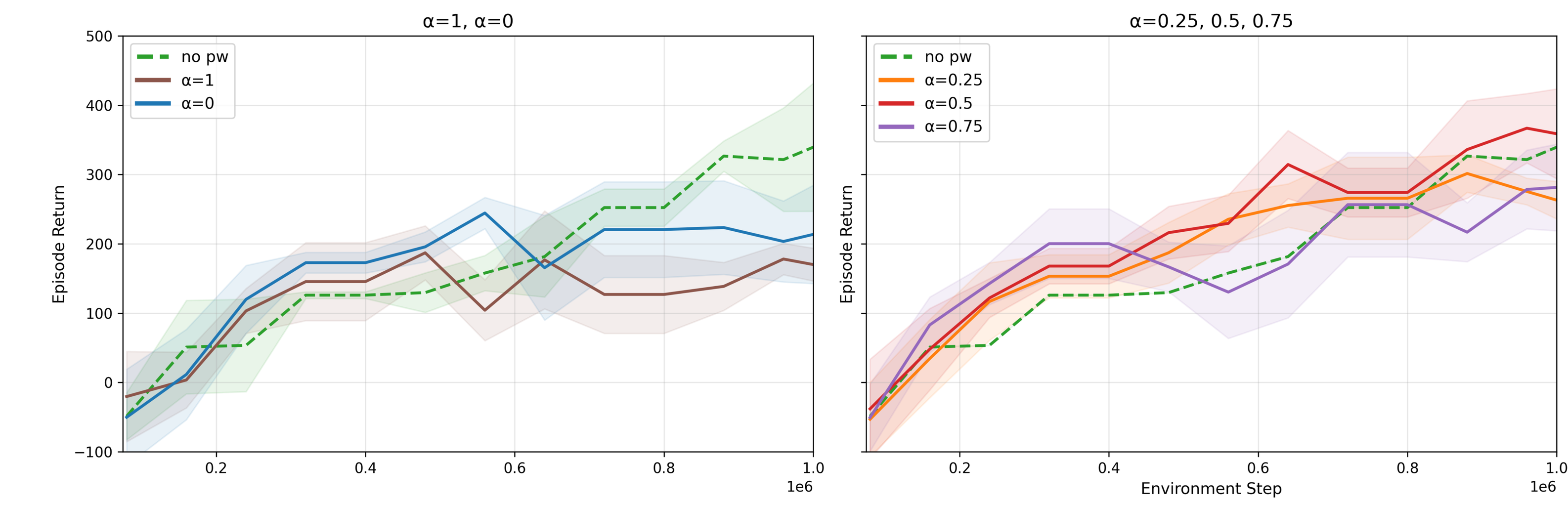

Figure 4. **Episode return across different α settings.** Shaded regions show standard error over 5 random seeds. The dotted line shows the baseline without progressive widening ($K = 20$). Progressive widening with $\alpha = 0.5$ achieves the best performance, outperforming both extreme values and the baseline.

Effect of Progressive Widening

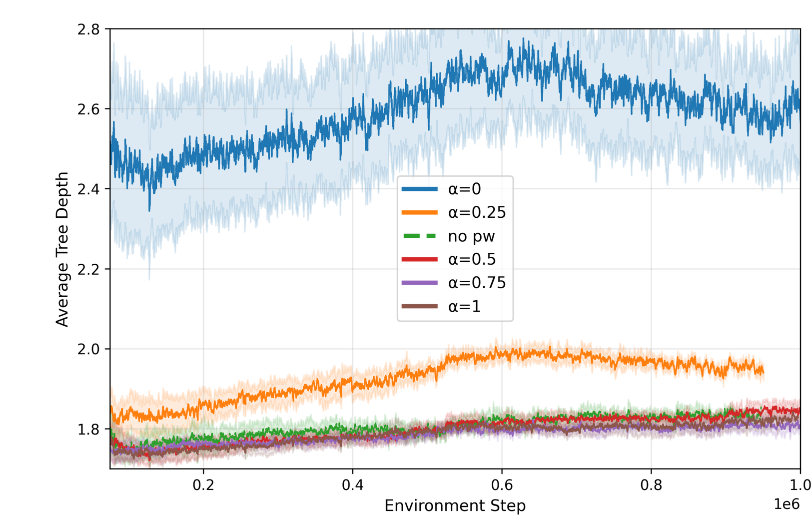

Figure 5. Episode return across different α settings. Shaded regions show standard error over 5 random seeds. The dotted line shows the baseline without progressive widening ($K = 20$). Progressive widening with $\alpha = 0.5$ achieves the best performance, outperforming both extreme values and the baseline.

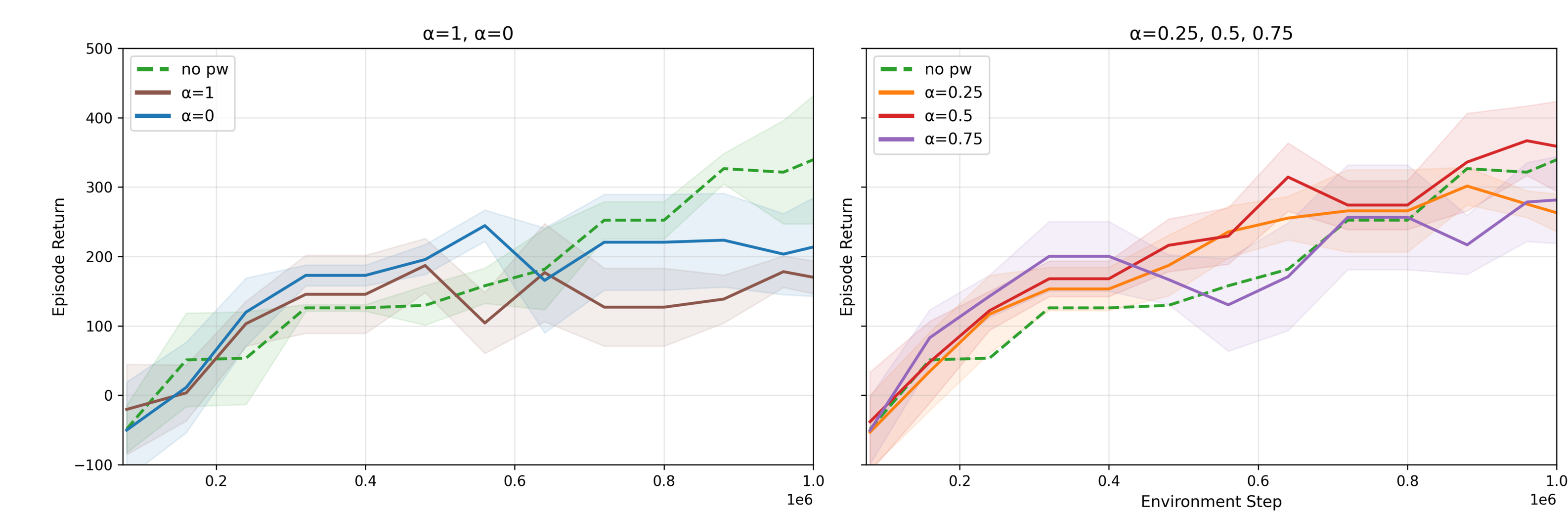

Figure 6. **Episode return across different α settings.** Shaded regions show standard error over 5 random seeds. The dotted line shows the baseline without progressive widening ($K = 20$). Progressive widening with $\alpha = 0.5$ achieves the best performance, outperforming both extreme values and the baseline.

Conclusion

- **Contributions:** First open-source JAX Sampled MuZero implementation; temperature modulation shows no improvement; progressive widening with proper α narrowly outperforms baseline
- **Limitations:** Testing limited to 1M steps and single environment; hyperparameter tuning may improve results
- **Future Work:** Test discretized policies, analyze branch depth distributions, implement Voronoi abstraction, extend to stochastic environments

References

[1] A. Couëtoux, J.-B. Hoock, N. Sokolovska, O. Teytaud, and N. Bonnard. Continuous Upper Confidence Trees. In *LION'11: Proceedings of the 5th International Conference on Learning and Intelligent OptimizatioN*, page TBA, Italy, Jan. 2011.

[2] T. Hubert, J. Schrittwieser, I. Antonoglou, M. Barekatain, S. Schmitt, and D. Silver. Learning and planning in complex action spaces, 2021.

[3] J. Schrittwieser, I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez, E. Lockhart, D. Hassabis, T. Graepel, T. Lillicrap, and D. Silver. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, Dec. 2020.

[4] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, and D. Hassabis. Mastering chess and shogi by self-play with a general reinforcement learning algorithm, 2017.

Author: **Vaclav Kubon** | Responsible professor: **Frans A. Oliehoek** | Supervisor: **Jinke He**

TU Delft