A Tree-based Scalable Behavior Task Planning Framework for Autonomous Systems

Mingyu Shin[1], Soyi Jung[2]
[1]Dept. Artificial Intelligence Convergence Network, Ajou University, Suwon, 16499, South Korea
[2]Dept. Electrical and Computer Engineering, Ajou University, Suwon, 16499, South Korea
{saycode99, sjung}@ajou.ac.kr

***Abstract*—Behavior trees (BTs) provide a hierarchical control architecture for autonomous systems, offering modularity and reactive behavior capabilities. However, traditional Q-learning behavior trees (QL-BT) rely on tabular representations, which face critical scalability challenges. This research presents a tree-based scalable behavior task planning framework that addresses these limitations by combining gradient-boosting trees (GBT) for function approximation and prioritized experience replay mechanisms. This integration leverages GBT's generalization capabilities and experience replay's stability benefits, achieving superior performance with reduced training requirements. The framework demonstrates substantial improvements in three key metrics: state-space exploration increases by 7.6 %, reward performance improves by 17.8 %, and computational performance improves by 110 %.**

***Index Terms*—Behavior Trees, Task Planning, Q-learning, Gradient Boosting Trees, Experience Replay, Function Approximation**

I. Introduction

Behavior trees (BTs) provide a hierarchical control architecture for autonomous systems, offering modularity and reactive behavior capabilities. Despite these advantages, conventional BTs require manual specification of transition conditions, which limits their adaptability to dynamic environments.

Q-learning behavior trees (QL-BT) integrate reinforcement learning to automate condition generation and action prioritization, enhancing environmental responsiveness [1]. However, current QL-BT implementations rely on tabular Q-learning, which faces critical scalability challenges. These methods require explicit storage of all state-action pairs, resulting in exponential memory growth with increasing state-space dimensionality. Furthermore, tabular approaches cannot generalize to unobserved states or exploit similarities between related states due to the independent treatment of each state-action pair [2].

This research presents a scalable QL-BT framework that overcomes these limitations by combining gradient-boosting trees (GBT) for function approximation with prioritized experience replay [3], [4]. The GBT approach builds an ensemble of decision trees to approximate Q-values without explicit Q-table storage, while experience replay enables efficient learning from historical transitions. This integration leverages GBT's generalization capabilities and experience replay's stability benefits, achieving superior performance with reduced training requirements compared to conventional Q-learning approaches.

II. Proposed Method

GBT-based Q-value Function Approximation. The framework approximates Q-value functions using gradient-boosting trees (GBT), replacing traditional tabular representations. The Q-value function is approximated as

$$Q(s,a) \approx F_K(s,a) = F_0 + \sum_{k=1}^{K} \varepsilon h_k(s,a), \tag{1}$$

where F_K represents the ensemble model after K boosting iterations, ε denotes the learning rate, F_0 is the initial estimate and h_k represents individual regression trees that partition the state-action space.

The GBT model is optimized by minimizing the squared loss function:

$$\mathscr{L}_{\text{GBT}} = \mathbb{E}_{(s,a)\sim\mathscr{D}}\left[(Q_{\text{target}}(s,a) - F_K(s,a))^2\right], \tag{2}$$

with target Q-values defined by the Bellman optimality equation:

$$Q_{\text{target}}(s,a) = r + \gamma \max_{a'} Q(s',a'). \tag{3}$$

This approach provides implicit regularization through boosting, while maintaining interpretability through tree-based architecture. Hierarchical state-space partitioning enables efficient processing of heterogeneous features and generalization to unobserved states.

Prioritized Experience Replay Integration. Prioritized experience replay improves learning stability and sample efficiency. Each transition receives a priority score based on the temporal difference (TD) error magnitude:

$$p_i = (|\delta_i| + \varepsilon_p)^{\alpha}, \tag{4}$$

where the TD error measures prediction discrepancy:

$$\delta_i = r_t + \gamma \max_{a'} Q(s_{t+1},a') - Q(s_t,a_t). \tag{5}$$

The sampling probability follows a normalized priority distribution:

$$P(i) = \frac{p_i}{\sum_k p_k}. \tag{6}$$

Importance sampling weights correct for non-uniform sampling bias:

$$w_i = \left(\frac{1}{N \cdot P(i)}\right)^{\beta}, \tag{7}$$

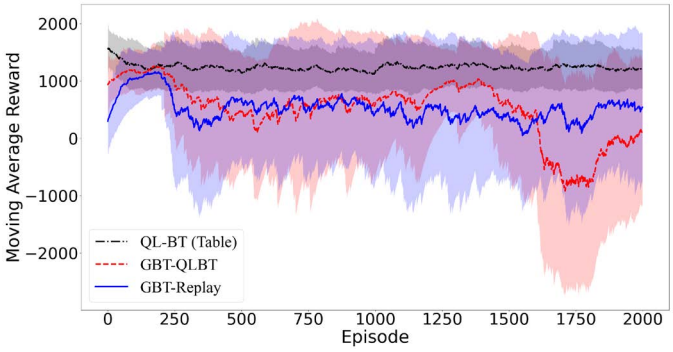

Fig. 1: Learning curves comparing QL-BT (Table), GBT-QLBT, and GBT-Replay models over 2,000 training episodes.

where N is the buffer capacity and β anneals linearly from 0 to 1 during training. This prioritization focuses computational resources on high-error transitions, accelerating convergence in sparse-reward environments.

Q-condition Generation and Tree Restructuring. Q-values are transformed into Q-condition nodes to replace manual conditions in behavior trees. For each action a, states with high Q-values form the condition set:

$$\mathscr{C}_a = \{s \in \mathscr{S} : Q(s,a) > \text{Percentile}_p(Q(\cdot,a))\}, \tag{8}$$

where the percentile threshold p (typically 70-90%) balances specificity and coverage.

The behavior tree undergoes structural optimization by reordering action nodes based on maximum Q-values:

$$\text{Priority}(\text{action}_i) = \max_{s \in \mathscr{C}_{a_i}} Q(s,a_i). \tag{9}$$

This reorganization prioritizes high-return actions during tree traversal, optimizing decision-making efficiency while preserving the interpretable hierarchical structure required for practical deployment.

III. Performance Evaluation

Experimental Configuration. The framework was evaluated over 2,000 training episodes in a 20 × 40 grid-based excavation environment. Three implementations were compared: traditional tabular Q-learning (QL-BT), GBT function approximation without experience replay (GBT-QLBT), and the proposed GBT with prioritized experience replay (GBT-Replay).

Learning Dynamics Analysis. Fig. 1 illustrates the distinctive convergence characteristics of the three approaches. The tabular QL-BT rapidly converged to a stable reward level of approximately 1,200. GBT-based implementations showed greater variance, and GBT-QLBT experienced performance degradation between episodes 1,500-1,800 due to overfitting during periodic retraining. GBT-Replay demonstrated superior stability through experience replay, effectively preventing catastrophic forgetting.

Performance Analysis. Table I demonstrates the synergistic advantages of function approximation and experience replay integration. GBT-QLBT explored 7.6 % more states than tabular approaches (732 versus 680), demonstrating enhanced exploration through generalization. However, its positive Q-ratio decreased to 64.5 %, indicating value propagation challenges without experience replay.

TABLE I: Comparative Performance Metrics

Metric	QL-BT	GBT-QLBT	GBT-Replay
State space coverage	680	732	725
Positive Q-ratio (%)	71.6	64.5	74.2
Mean Q-value	115.40	131.45	148.21
Average reward	560.80	551.33	660.74
Decision latency (ms)	7.15	7.28	3.40
Throughput (dec/s)	140.5	140.5	294.5

GBT-Replay achieved optimal performance across all metrics: highest positive Q-ratio (74.2 %), maximum mean Q-value (148.21) and best average reward (660.74). This represents a 17.8 % reward improvement over QL-BT and 19.8 % over GBT-QLBT. Computational efficiency improved significantly, with GBT-Replay achieving a decision latency of 3.40 ms - a 52.4 % reduction compared to tabular implementations. This yielded 294.5 decisions per second, doubling the throughput of alternative methods and enabling real-time deployment.

IV. Conclusion

This research has presented a scalable QL-BT framework that addresses the computational constraints of traditional tabular Q-learning through the integration of GBT-based function approximation and prioritized experience replay mechanisms. The proposed methodology demonstrates substantial improvements across multiple dimensions: state-space exploration (7.6% increase), learning performance (17.8% reward improvement), and computational efficiency (110% throughput enhancement). The framework preserves the interpretability inherent in behavior trees while achieving the computational scalability necessary for deployment in high-dimensional state spaces.

Acknowledgment

This work was supported by the Technology Innovation Program (1415187715, Development of AI learning platform for intelligent excavators based on expert work data) funded By the Ministry of Trade, Industry & Energy(MOTIE, Korea)

References

[1] R. Dey and C. Child, "QL-BT: Enhancing behaviour tree design and implementation with Q-learning," in *2013 IEEE Conference on Computational Intelligence in Games (CIG)*. Niagara Falls, ON, Canada: IEEE, August 2013, pp. 1–8.

[2] H. Lee, S. Jung, and S. Park, "Situation-aware deep reinforcement learning for autonomous nonlinear mobility control in cyber-physical loitering munition systems," *IEEE/KICS Journal of Communications and Networks*, vol. 27, no. 1, pp. 10–22, Feb. 2025.

[3] B. Fuhrer, C. Tessler, and G. Dalal, "Gradient boosting reinforcement learning," 2025. [Online]. Available: https://arxiv.org/abs/2407.08250

[4] J. Jang, J. Kim, J. Kim, and S. Jung, "Joint interference approximation and guard-band management for spectrum-efficient integrated ntn–tn networks," *IEEE Internet of Things Journal*, vol. 12, no. 15, pp. 32 220–32 236, Aug 2025.