Scalable On-Policy Reinforcement Learning via Adaptive Batch Scaling

Jongchan Park*

Hyundai Motor Company

{jcpark11@hyundai.com}

Research Question.

- Why most of reinforcement learning (RL) employ modest batch sizes?
  - Many RL studies adopt **small batch sizes** compared to standard supervised learning (SL).
  - Several studies report that **increasing the batch size** often leads to **diminishing returns** or even **performance degradation** in **RL**[1, 2].

[1] McCandlish, S., Kaplan, J., Amodei, D., and Team, O. D. An empirical model of large-batch training. arXiv preprint arXiv:1812.06162, 2018.
[2] Obando Ceron, J., Bellemare, M., and Castro, P. S. Small batch deep reinforcement learning. Advances in Neural Information Processing Systems, 36:26003–26024, 2023.

Research Question.

- Why most of reinforcement learnings (RL) employ modest batch size?
  - This phenomenon extends even to **scaled RL models**, such as IMPALA[1], SHARSA[2]; they utilize small batch sizes of 32 and 256 respectively.
  - Furthermore, in RL stages for **Large Language Model (LLM)** training, they also utilize notably smaller batch sizes compared to the massive scales employed during SL phases[3, 4].

[1] Espeholt, Lasse, et al. "Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures." International conference on machine learning. PMLR, 2018.
[2] Park, Seohong, et al. "Horizon reduction makes rl scalable." Advances in Neural Information Processing Systems 38 (2026): 8350-8389.
[3] Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. Advances in neural information processing systems, 36: 53728–53741, 2023.
[4] Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. arXiv preprint arXiv:2501.12948, 2025.

Our Hypothesis.

- We suppose that this stems from the **inherent non-stationary of RL**.
  - An **evolving policy** and the resulting **data distribution shifts** during training create this non-stationary.
  - This non-stationary is **particularly acute during early learning stages**, where rapid policy shifts exacerbate distributional instability.
  - To mitigate these effects, **employing modest batch sizes is often preferred**, as it allows the agent to adapt more fluidly to the non-stationary data stream[1].

[1] Obando Ceron, J., Bellemare, M., and Castro, P. S. Small batch deep reinforcement learning. Advances in Neural Information Processing Systems, 36:26003–26024, 2023.

Our Hypothesis.

- We suppose that this stems from the **inherent non-stationary of RL**.
- We also assume that the **policy and environment will stabilize in late learning phase** which we call '**near-stationary**' phase.

Our Insight.

- We suppose that this stems from the **inherent non-stationary of RL**.
- We also assume that the **policy and environment will stabilize in late learning phase** which we call '**near-stationary**' phase.

→ **Can we leverage the benefits of large batch sizes by increasing them during this near-stationary phase?**

Behavioral Divergence.

- We quantify non-stationarity by comparing consecutive policy updates.

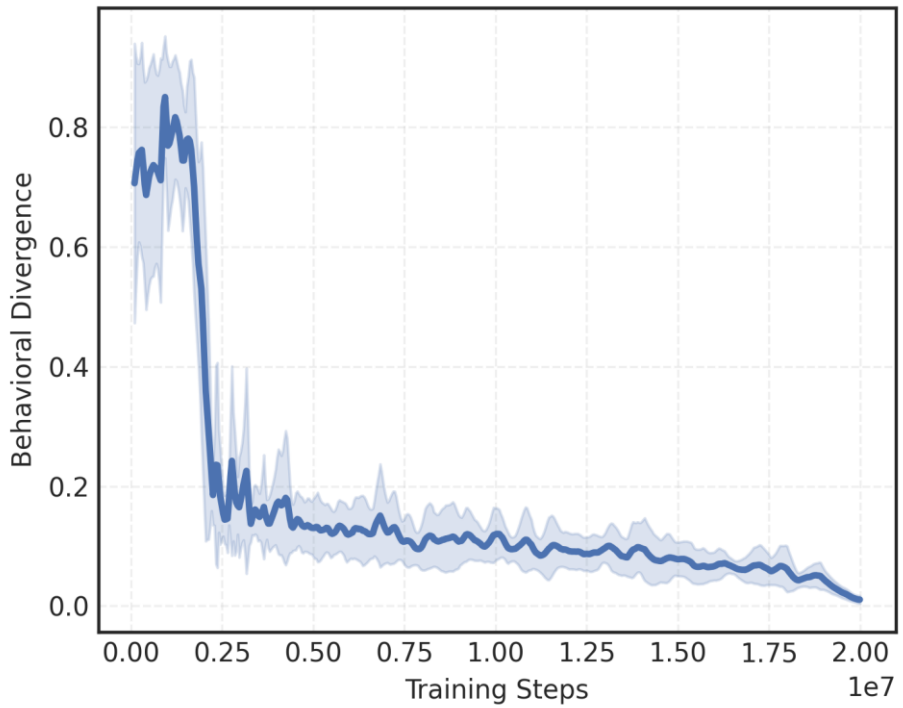
$$\delta_\pi(\theta', \theta) := \frac{1}{M} \sum_{i=1}^{M} \mathbb{I}\left[\pi_{\theta'}(s_i) \neq \pi_\theta(s_i)\right], \; s_i \sim \mathcal{B}_{ref}$$

  - δ_π: Behavioral Divergence (BD).
  - $\mathcal{B}_{ref}$: Replay buffer to calculate BD.
  - M: Size of $\mathcal{B}_{ref}$.
  - $\pi_{\theta'}, \pi_\theta$: updated and previous policies respectively.

Behavioral Divergence.

- Figure 1 shows that Behavioral Divergence shrinks as training progress.
- This indicates that the **non-stationarity decreases** over training steps.

Figure 1



Adaptive Batch Scaling (ABS).

- Utilize **small batches** during the **non-stationary early training phase**

  → To maximize gradient update frequency and preserve **plasticity**

- Transition to **larger batches** in the **near-stationary later phase**

  → To ensure **stable convergence**

Adaptive Batch Scaling (ABS).

- ABS dynamically scales batch sizes $|\mathcal{B}|$ by adjusting the rollout length L_{adapt}:

$$|\mathcal{B}| = E \cdot L_{\text{adapt}}$$

- L_{adapt} in response to the measured Behavioral Divergence δ_π

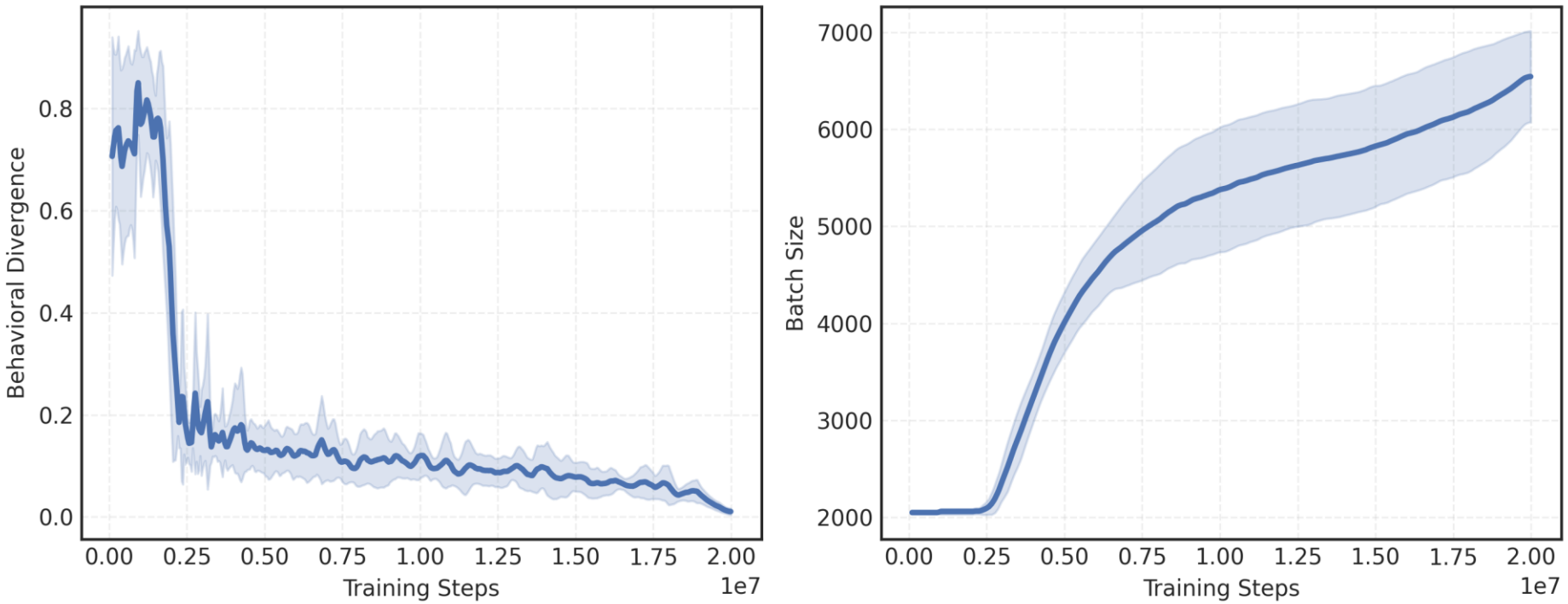
$$L_{\text{adapt}} = L_{\max} - \alpha \cdot (L_{\max} - L_{\min}) \qquad \alpha = \frac{\log(\delta_{\pi,\text{cliped}}/\delta_{\min})}{\log(\delta_{\max}/\delta_{\min})}$$

  - $\delta_{\pi,\text{cliped}} = \text{clip}(\delta_\pi, \delta_{\min}, \delta_{\max})$
  - $[\delta_{\min}, \delta_{\max}]$ represents the sensitivity range of the policy change rate

Adaptive Batch Scaling (ABS).

- Figure 2 shows that the batch size of ABS increases inversely with Behavioral Divergence.

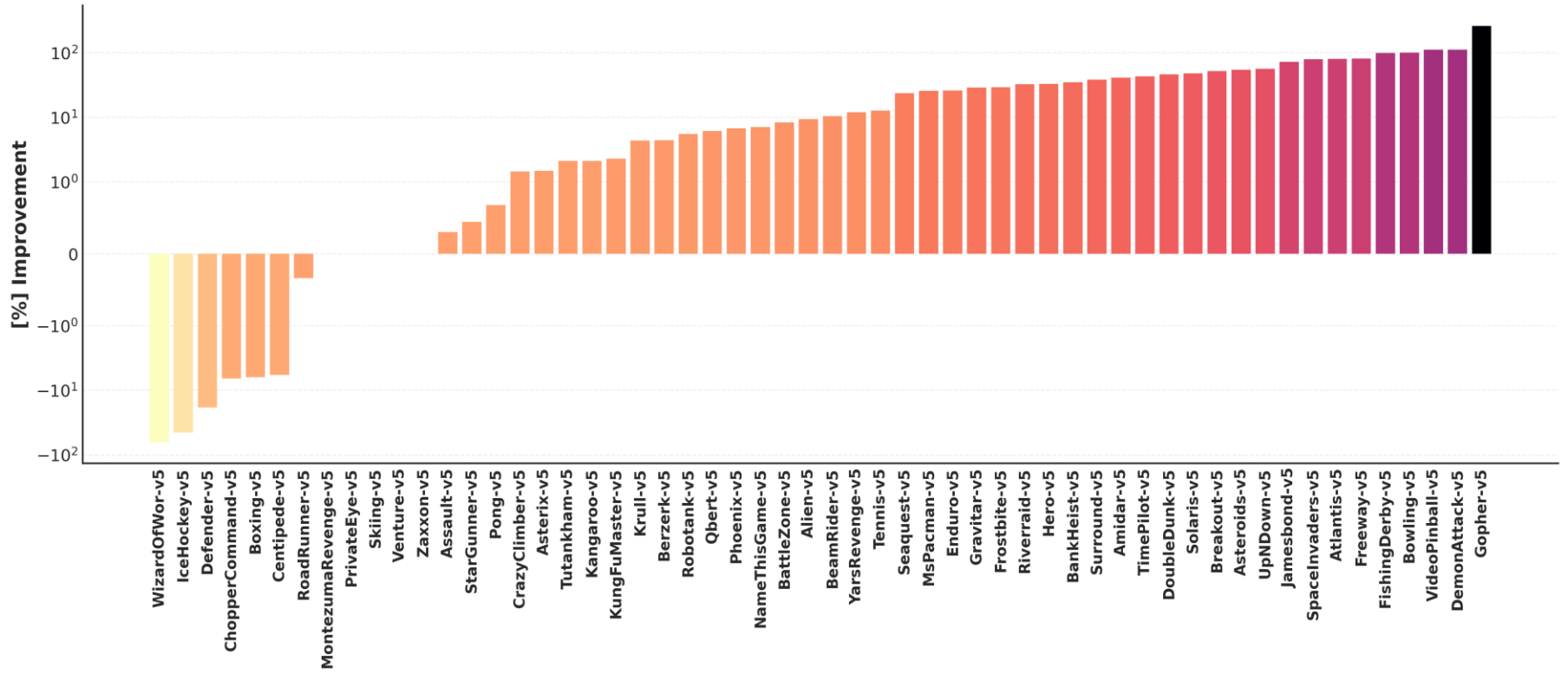
Figure 2



Experimental Results.

- Figure 3 demonstrates that our ABS increases PQN[1] performance across most of the Atari-57 benchmark.

Figure 3

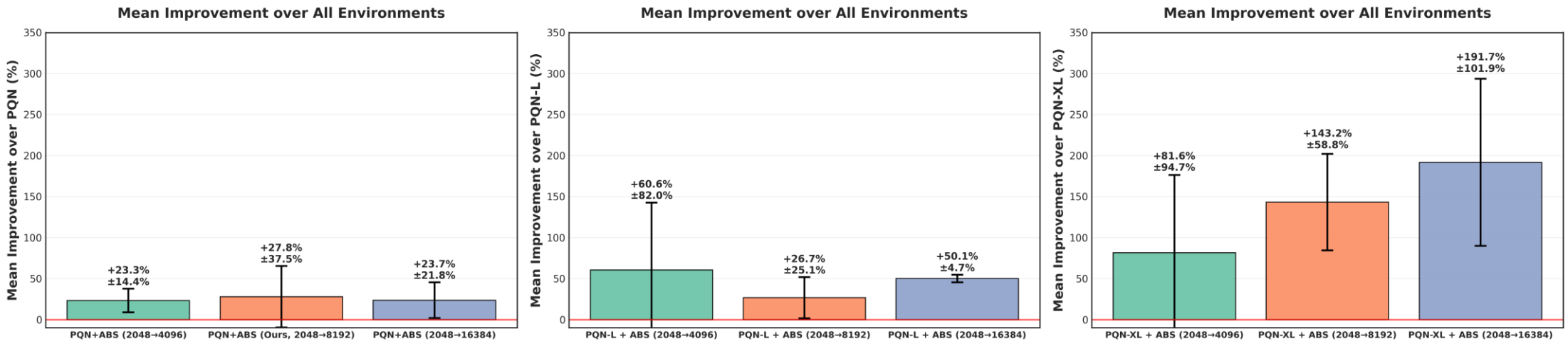


[1] Gallici, M., Fellows, M., Ellis, B., Pou, B., Masmitja, I., Foerster, J. N., and Martin, M. Simplifying deep temporal difference learning. arXiv preprint arXiv:2407.04811, 2024.

Experimental Results.

- Figure 4 shows that the mean improvement over each baseline (PQN, PQN-L[1], PQN-XL[1]) across various maximum batch sizes on Atari-10[2]

Figure 4



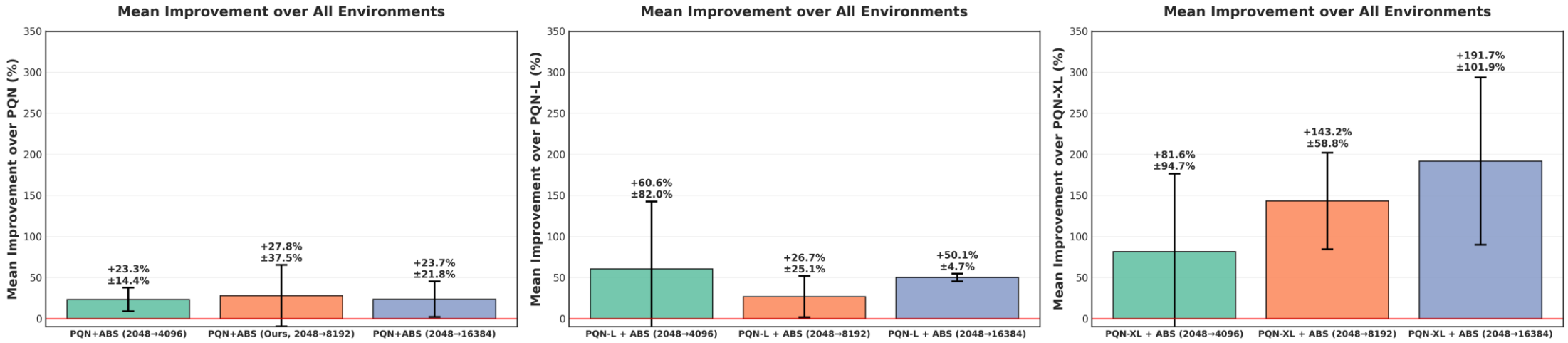
[1] Castanyer, R. C., Obando-Ceron, J., Li, L., Bacon, P.-L., Berseth, G., Courville, A., and Castro, P. S. Stable gradients for stable learning at scale in deep reinforcement learning. arXiv preprint arXiv:2506.15544, 2025.

[2] Aitchison, M., Sweetser, P., and Hutter, M. Atari-5: Distilling the arcade learning environment down to five games. In International Conference on Machine Learning, pp. 421–438. PMLR, 2023.

Experimental Results.

- As **model capacity** and **maximum batch size increase**, the **improvements** yielded by ABS become **more pronounced**.
  - For PQN-XL (the largest model), ABS achieves the most substantial gains when scaling the batch size from 2,048 to 16,384 samples.

Figure 4



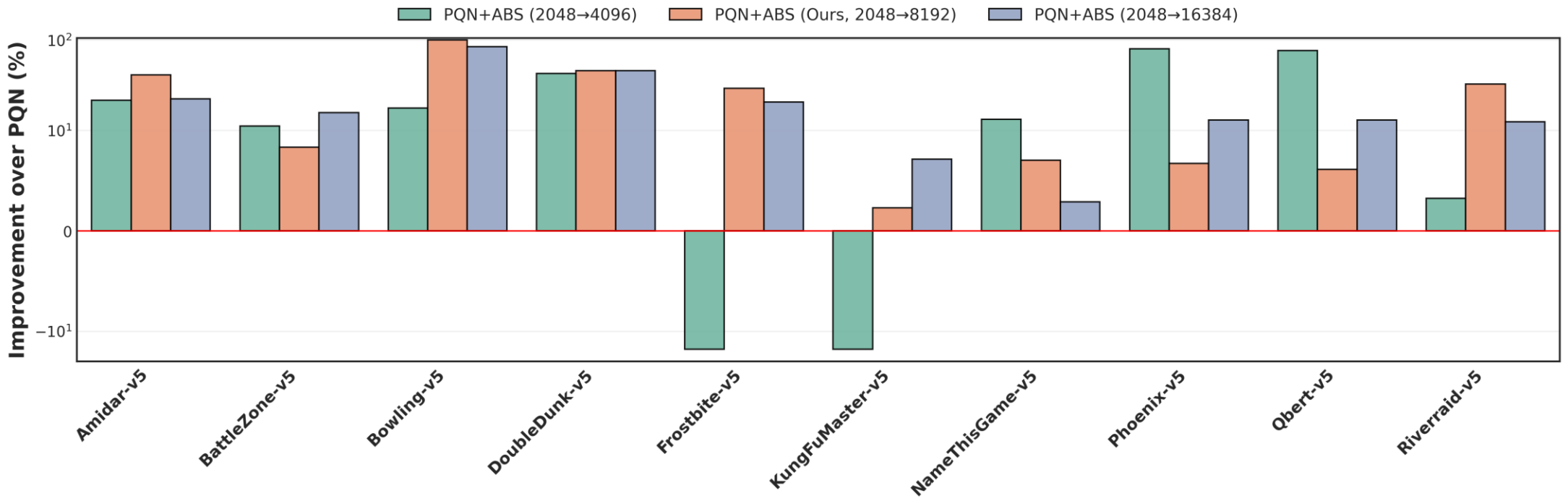
Conclusion.

- **Core Insight**: RL's inherent non-stationarity stabilizes in the later phase, allowing the effective use of larger batch sizes.
- **Proposed ABS**: Automatically scales up the batch size as behavioral divergence shrinks, avoiding early-stage performance drops.
- **Scalable RL Contribution**: Maximizes training efficiency for large-scale RL models, enabling more effective and scalable training.

Appendix.

- Figure 5 shows the detailed improvement of PQN+ABS in Figure 4 (Left) on the Atari-10 environments

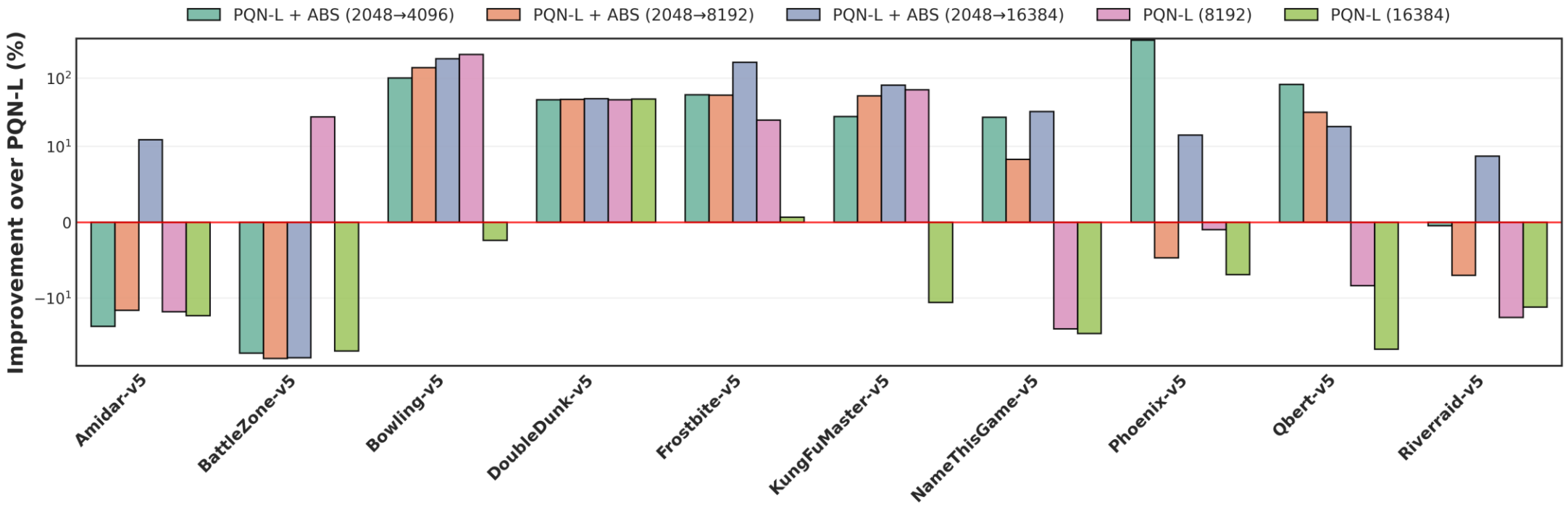
Figure 5



Appendix.

- Figure 6 shows the detailed improvement of PQN-L+ABS in Figure 4 (Center) on the Atari-10 environments

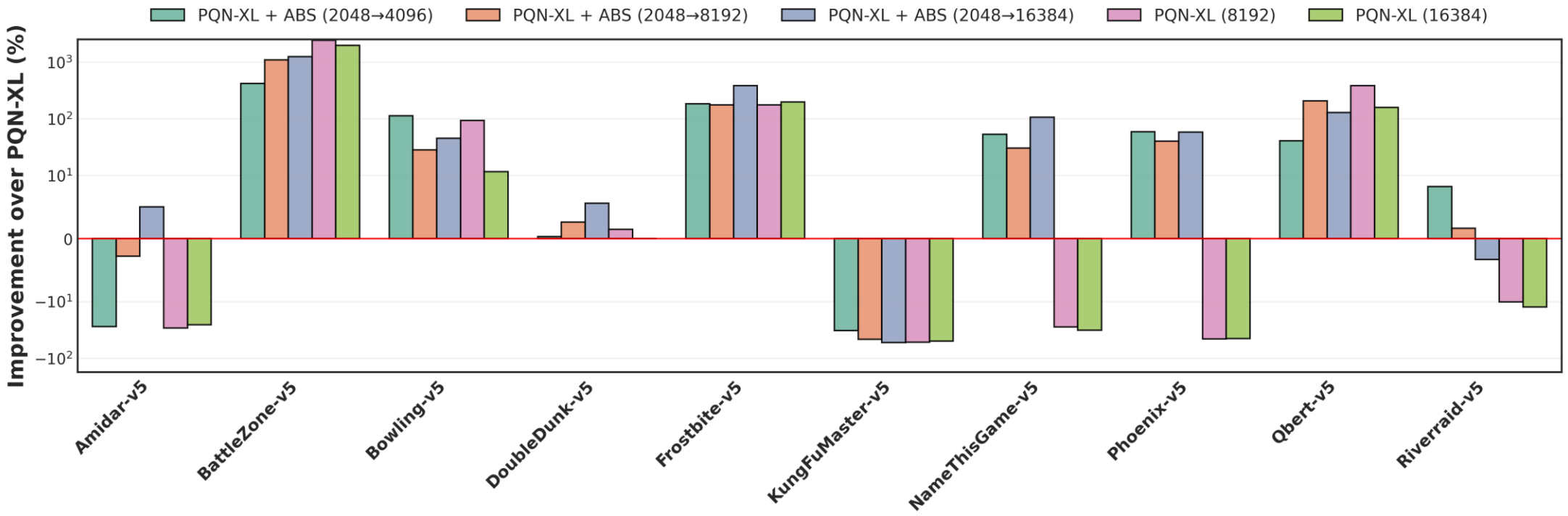
Figure 6



Appendix.

- Figure 7 shows the detailed improvement of PQN-XL+ABS in Figure 4 (Right) on the Atari-10 environments

Figure 7



Appendix.

Algorithm 1 PQN with Adaptive Batch Scaling (ABS)

Input: Initial parameters θ, environment count E, update frequency K, thresholds $\delta_{\min}, \delta_{\max}$, bounds $L_{\min}, L_{\max}$.

while $t < T_{\text{total}}$ **do**
  If $t \pmod K == 0$, set $\theta_{\text{old}} \leftarrow \theta$.

  Step 1: Data Collection
  Collect trajectories for L_{adapt} steps using E environments and policy π_θ.
  $t = t + (E \times L_{\text{adapt}})$

  Step 2: Policy Update
  for $epoch = 1$ **to** N_{epochs} **do**
    Sample mini-batches from the collected trajectories.
    Update θ by minimizing PQN loss.
  end for

  Step 3: Adaptive Scaling
  if $t \pmod K == 0$ **then**
    Sample reference batch $\mathcal{B}_{\text{ref}}$ of size M from the collected trajectories.
    Calculate behavioral divergence:
    $\delta_\pi = \frac{1}{M} \sum \mathbb{I}[\pi_\theta(s) \neq \pi_{\theta_{\text{old}}}(s)], \quad s \sim \mathcal{B}_{\text{ref}}$
    Calculate L'_{adapt} based on δ_π (Eq. 4).
    Smooth update: $L_{\text{adapt}} \leftarrow (1-\alpha)L_{\text{adapt}} + \alpha L'_{\text{adapt}}$
  end if
end while

PQN w/ ABS

Parameter	Value
General Hyperparameters	
Total Timesteps	2×10^7
Num. Environments	128
Frame Stack	4
Sticky Action Probability	0
Life Information	False
Learning Rate	2.5×10^{-4}
Base Steps per Rollout ($L_{\min}$)	32
Discount Factor (γ)	0.99
Mini-batches	4
Update Epochs (N_{epochs})	2
Max Grad Norm	10.0
Exploration ($\epsilon_{\text{start}} \rightarrow \epsilon_{\text{end}}$)	$1.0 \rightarrow 0.001$
Exploration Fraction	0.10
Q-Learning λ	0.65
Adaptive Rollout Hyperparameters	
Adapt Rollout	True
Rollout Range ($L_{\min}, L_{\max}$)	[16, 64]
Adapt Frequency (K)	50 iterations
Policy Change Thresholds ($\delta_{\min}, \delta_{\max}$)	[0.05, 0.95]
Schedule Type	log

PQN-L/XL w/ ABS

Parameter	Value
General Hyperparameters	
Total Timesteps	2×10^7
Num. Environments	128
Frame Stack	4
Sticky Action Probability	0
Life Information	False
Learning Rate	2.5×10^{-4}
Anneal LR	False
Base Steps per Rollout ($L_{\min}$)	32
Discount Factor (γ)	0.99
Mini-batches	4
Update Epochs (N_{epochs})	2
Max Grad Norm	10.0
Exploration ($\epsilon_{\text{start}} \rightarrow \epsilon_{\text{end}}$)	$1.0 \rightarrow 0.001$
Exploration Fraction	0.10
Q-Learning λ	0.65
Multi-Skip Network Architecture	
Use Multi-Skip Residual MLP	True
MLP Hidden Size	512
MLP Layers	5
Layer Normalization	True
Activation Function	ReLU
CNN Channels	(64, 128, 128)
Adaptive Rollout Settings	
Adapt Rollout	True
Rollout Range ($L_{\min}, L_{\max}$)	[16, 128]
Adapt Frequency (K)	50 iterations
Policy Change Thresholds ($\delta_{\min}, \delta_{\max}$)	[0.05, 0.95]
Schedule Type	log