Compute-Optimal Scaling for Value-Based Deep RL

Preston Fu*, Oleh Rybkin*, Zhiyuan Zhou, Michal Nauman, Pieter Abbeel, Sergey Levine, Aviral Kumar

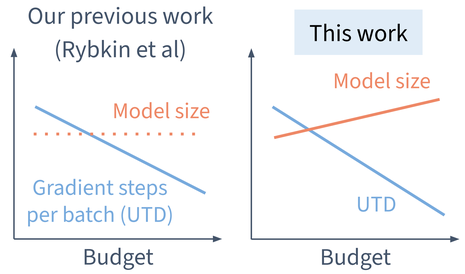
Berkeley UNIVERSITY OF CALIFORNIA · Carnegie Mellon University · UNIVERSITY OF WARSAW

Motivation and Problem Statement

Q1: RL is sensitive to hyperparameters. How to set batch size for large-scale runs?

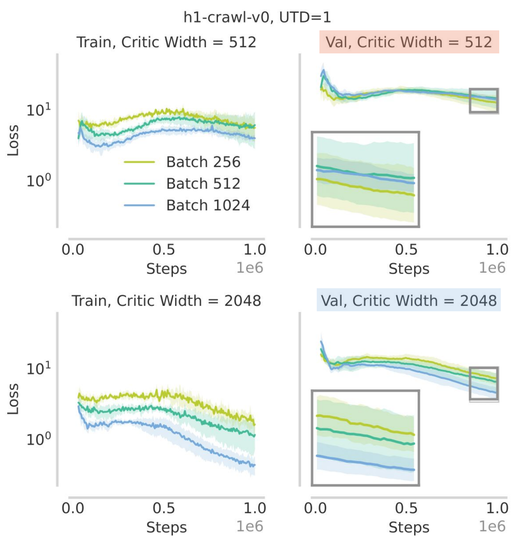
Q2: To achieve performance level J, RL requires a combination of *data* $\mathcal{D}$ and *compute* $\mathcal{C}$. What is their tradeoff?

Q3: I have a requirement on the total *budget*, a combination $\mathcal{F} = \mathcal{C} + \delta \cdot \mathcal{D}$ of data and compute. What algorithm configuration maximizes performance given this budget?



Q0: How does model size modulate batch size?

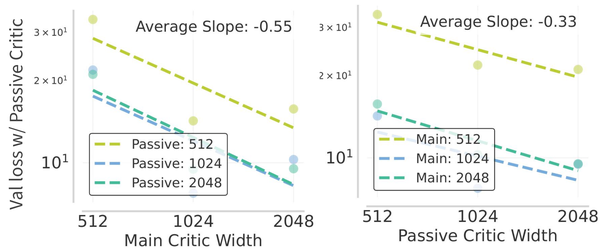
Our previous work found the best batch size should **decrease with the updates-to-data ratio** (UTD; gradient steps per batch) to counteract overfitting effects. What if we fix UTD and vary model size?



Training TD-error improves with batch size.
For **small** models, val **worsens** with batch size.
For **large** models, val **improves** with batch size.
Why does this happen?

This discrepancy is due to poor **generalization of TD-targets** produced by smaller models.

Idea: Train a "passive critic" alongside the main critic that regresses to TD-targets. This decouples the main critic's capacity from TD-target quality.



Validation TD-error under the passive critic is more effectively reduced by improving TD-targets!

TD-overfitting: Small models benefit from smaller batch sizes, which result in noisy gradient updates. Large models produce high-quality TD-targets and can benefit from low-variance updates.

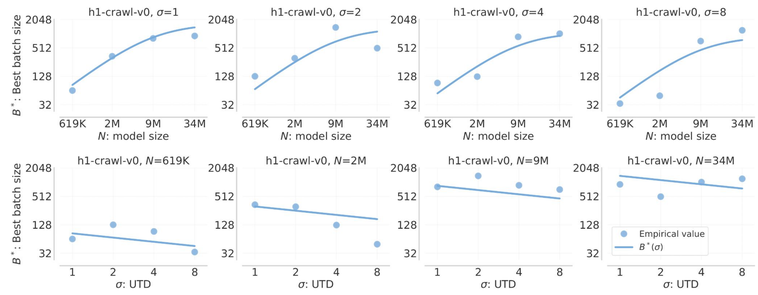
Q1: What is the best batch size?

From our previous work, the best batch size **decreases with UTD**.

From TD-overfitting, the best batch size **increases with model size**.

Combining these independently (empirically good enough):

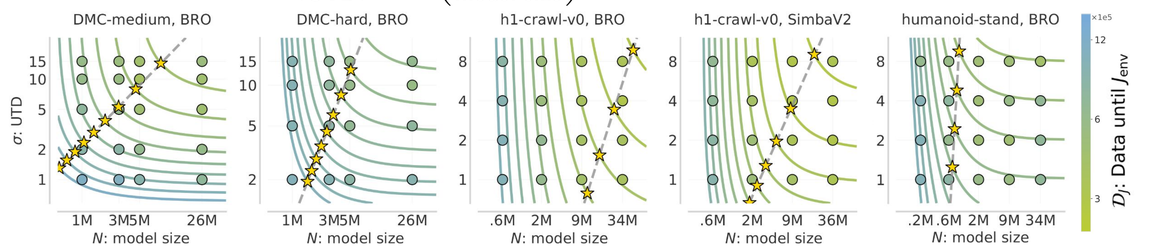
$$\frac{a_B}{\text{UTD}^{\alpha_B}} \cdot \frac{1}{1 + b_B \cdot (\text{model size})^{-\beta_B}}$$



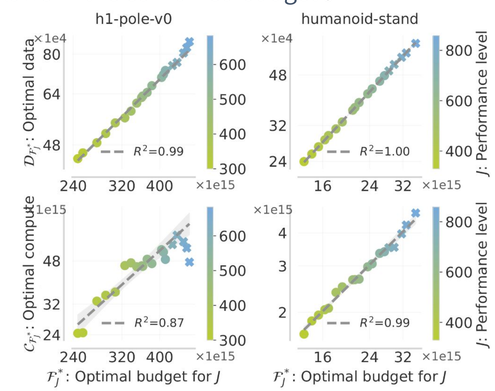
Q2, Q3: Configuring Algorithm

With the right batch size, data and compute are predictable functions of UTD and model size.

$$\mathcal{D}_J \approx \mathcal{D}_J^{\min} + \left(\frac{a_J}{\text{UTD}}\right)^{\alpha_J} + \left(\frac{b_J}{\text{model size}}\right)^{\beta_J} \qquad \mathcal{C}_J \approx k \cdot \text{UTD} \cdot \text{model size} \cdot \mathcal{D}_J$$



Q2 Optimal compute and data follow power laws in the combined budget $\mathcal{F}$.



Q3 The required precision in budget-optimal UTD and model size varies across tasks.