A Study of the Weighted Multi-step Loss Impact on the Predictive Error and the Return in MBRL

Abdelhakim Benechehab[1,2] Albert Thomas[1] Giuseppe Paolo[1] Maurizio Filippone[3] Balázs Kégl[1]

[1]Huawei Noah's Ark Lab [2]Department of Data Science, EURECOM [3]Statistics Program, KAUST

TL;DR

- Models face compounding errors and a distribution mismatch at test time
- The **Weighted Multi-Step loss** is a way to solve this problem
- Although it improves the predictive error, it doesn't necessarily lead to better policies

Problem Setup

Goal: given a dataset of real system trajectories, learn a parametric model of its transition function.

- Input $[s_t, a_t] \in \mathbb{R}^{d_s+d_a}$, target $s_{t+1} \in \mathbb{R}^{d_s}$,
- Training set of N trajectories $\mathcal{D} = \{(s_0^i, a_0^i, s_1^i, \ldots)\}_{i=1}^N$,
- Train a model $\hat{p}_{\boldsymbol{\theta}}: \mathbb{R}^{d_s+d_a} \to \mathbb{R}^{d_s}$ that minimizes the MSE loss (or NLL).

Single-step error

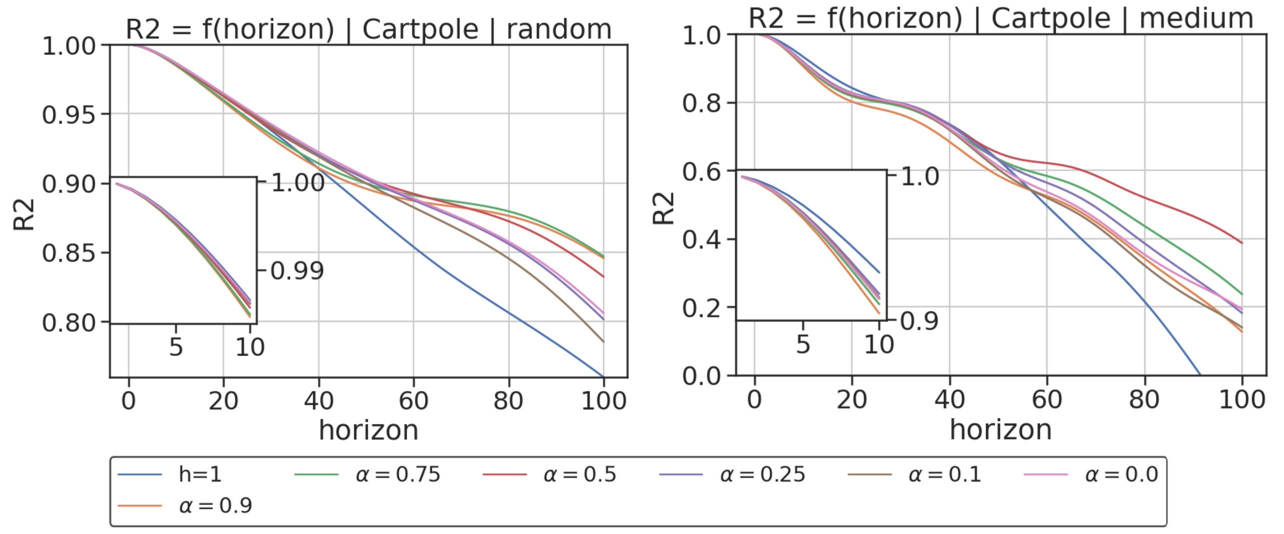
99% R2

Multi-step error

Please help, errors are compounding!

Compounding errors

- At training time: the model only sees single-step transitions $s_{t+1} \sim p_{true}(.|s_t, a_t)$,
- At test time: generate long rollouts recursively $\hat{s}_{t+j} \sim \hat{p}_\theta(.|\hat{s}_{t+j-1}, a_{t+j-1})$,
- **Distribution mismatch** training $s_t \sim p_{true}$, test $s_t \sim \hat{p}_\theta$,
- **Compounding errors** $\hat{p}_\theta(.|\hat{p}_\theta(.|\hat{s}_{t+j-2}, a_{t+j-2}), a_{t+j-1})$



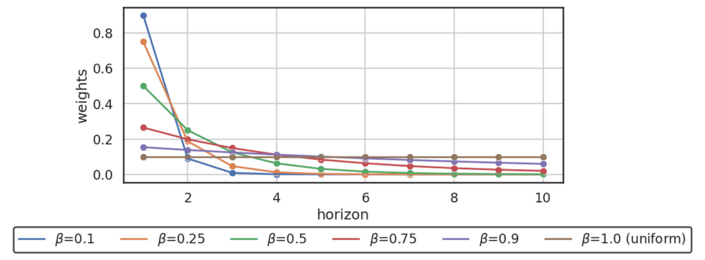
Solution: Weighted Multi-Step Loss

- horizon-dependent **weights** $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_h)$ with $\sum_{i=1}^h \alpha_i = 1$,
- a single-step **loss function** L (MSE),
- an initial state s_t, an action sequence $\mathbf{a}_\tau = \mathbf{a}_{t:t+h-1}$, and the real (ground truth) visited states $\mathbf{s}_\tau = \mathbf{s}_{t+1:t+h}$,
- we define the weighted multi-step loss of horizon h as:

$$L_{\boldsymbol{\alpha}}^h(\mathbf{s}_\tau, \hat{p}_\theta(s_t, \mathbf{a}_\tau)) = \sum_{j=1}^h \alpha_j L(s_{t+j}, \hat{p}_\theta^j(s_t, \mathbf{a}_{t:t+j-1}))$$

How to choose the weights $\boldsymbol{\alpha}$.

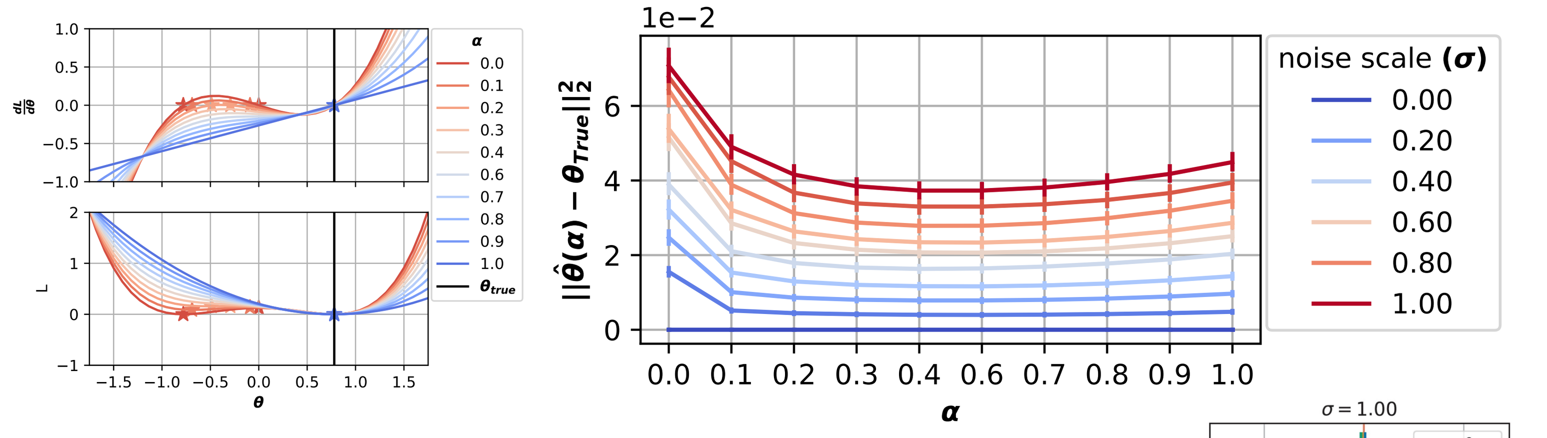
- **Uniform.** $\alpha_j = 1/h$ The simplest choice,
- **β-Decay.** $\alpha_j = \frac{1}{Z}\beta^j$ Inspired by the error growth profile
- **Learn.** $\alpha_j = learnable$ (Not well defined)
- **Proportional.** $\alpha_j \sim \frac{1}{L\left(s_{t+j}, \hat{p}_\theta^j(s_t, \mathbf{a}_{t:t+j})\right)}$ all terms are equally-important, regardless of the amplitude



Theoretical insights: uni-dimensional linear system

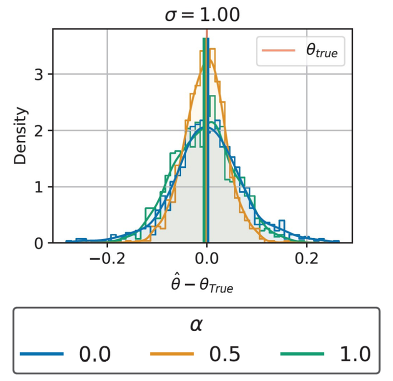
- **System.** $s_{t+1} = \theta_{true} \cdot s_t$ and $o_{t+1} = s_{t+1} + \epsilon_{t+1}$ with $\epsilon_{t+1} \sim \mathcal{N}(0, \sigma^2)$
- **Problem.** We study the minimizers $\hat{\theta}(\alpha) \in \arg\min_\theta L_\alpha(\mathbf{o}_\tau, \hat{p}_\theta(s_t))$ where

$$L_\alpha(\mathbf{o}_\tau, \hat{p}_\theta(s_t)) = \alpha(\theta s_t - o_{t+1})^2 + (1-\alpha)(\theta^2 s_t - o_{t+2})^2$$



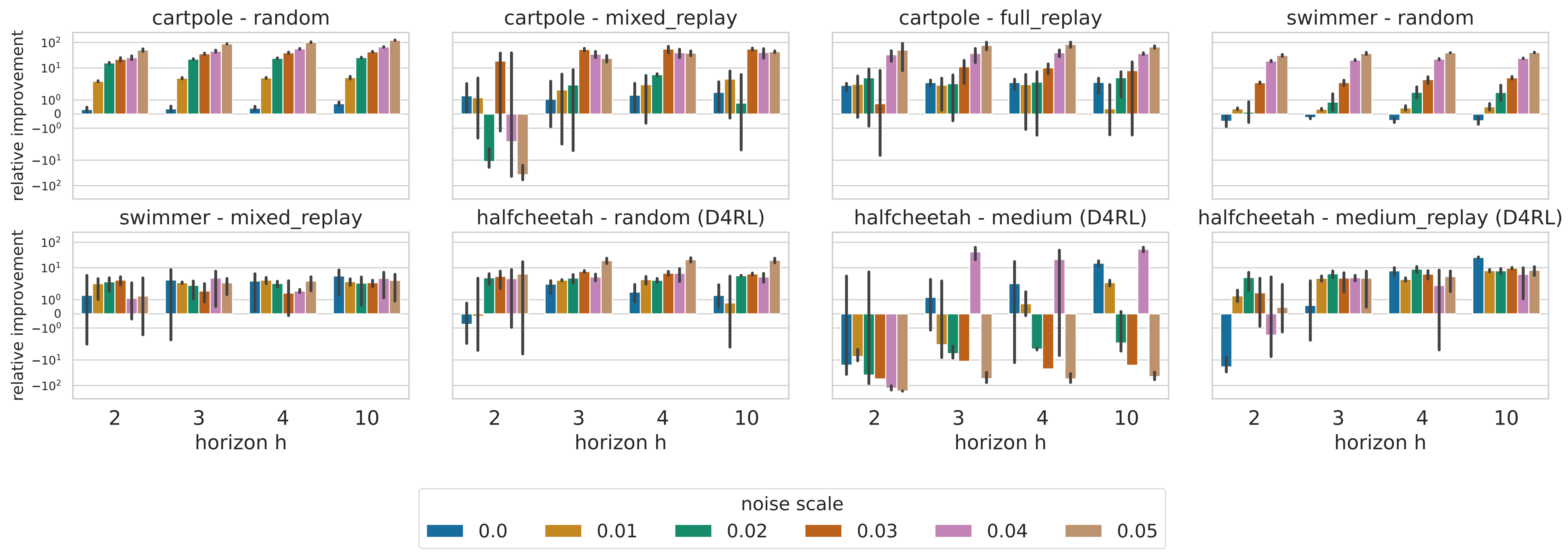
insights:

- $\alpha = \mathbf{1}.$ the minimizer is unbiased: $E_{\epsilon_{t+1} \sim \mathcal{N}(0,\sigma^2)}[\hat{\theta}_1] = \theta_{true}$
- $\alpha = \mathbf{0}.$ the minimizer is biased but has lower variance under some conditions
- $\alpha \in (\mathbf{0}, \mathbf{1}).$ provides the best bias-variance **tradeoff** empirically



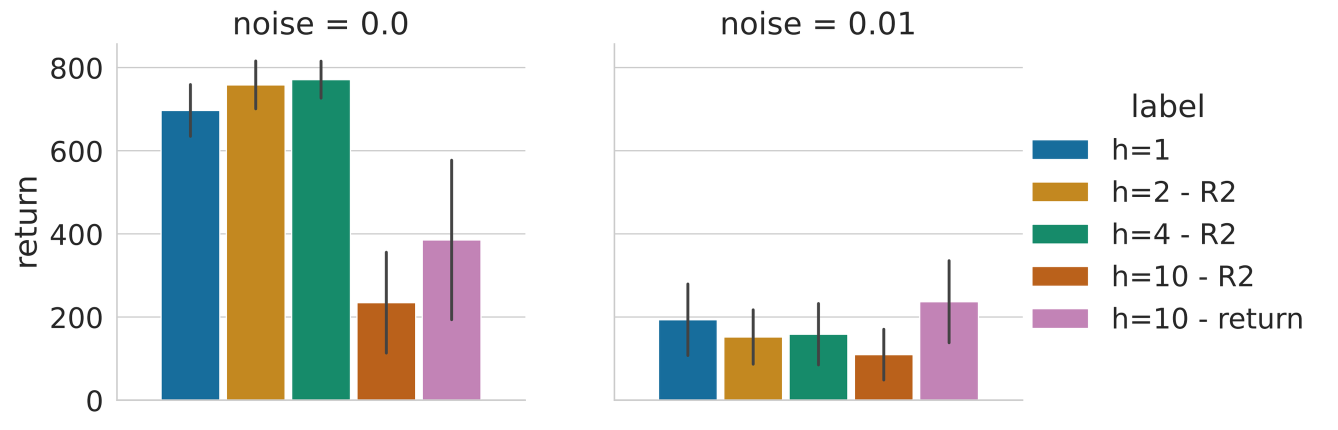
Experimental Results

- **setup.** noisy observations $o_t = s_t + \epsilon_t$ with $\epsilon_t \sim \mathcal{N}(0, \sigma^2)$
- **Metric.** aggregated R2 score $\overline{R2}(H) = \frac{1}{H}\sum_{h=1}^H \mathrm{R2}(h)$
- **Benchmark.** Environments (Cartpole swingup, Halfcheetah, Swimmer), Datasets (random, medium, replay)



Offline MBRL

- **agent.** *Dyna*-style using Soft Actor-Critic (SAC) a la MBPO
- $\mathbf{h} = \mathbf{1}.$ the baseline
- **h = h - R2.** we select the optimal β value in grid search based on the $R2$ metric
- **h = h - return.** we select the optimal β value in grid search based on the return of the agent
- **task.** Cartpole swing-up mixed replay dataset, with two levels of noise 0% and 1%



insights:

- We can have a small improvement over the baseline using the weighted multi-step loss
- Large values of the loss horizon h do not work in practice
- In the noisy variant, noise is probably too large to learn any meaningful policy
- **More experiments are needed to conclude**

Take Home Message

The **Weighted Multi-step loss** is useful to improve the **predictive error** down the horizon.
→ But is this a good metric for model selection in MBRL ?!

Want to Know More?

Main References

- **MBPO, Janner et al.** - Neurips 2019
  When to Trust Your Model: Model-Based Policy Optimization
- **Lambert et al.** - arXiv:2203.09637 2022
  Investigating Compounding Prediction Errors in Learned Dynamics Models
- **Benechehab et al.** - **ICBINB workshop at RLC 2024 (this work)**
  A Study of the Weighted Multi-step Loss Impact on the Predictive Error and the Return in MBRL

Correspondence to abdelhakim.benechehab@gmail.com