# Large Batch Experience Replay

Thibault Lahire, Matthieu Geist, Emmanuel Rachelson

ISAE-SUPAERO & Google Research

# Take-away messages

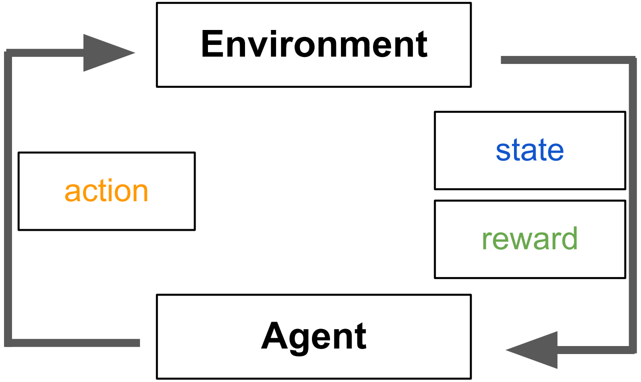
- Deep Reinforcement Learning and replay buffers : the DQN case
- Prioritized Experience Replay (PER) : samples transitions with high error
- PER is a heuristic for DQN: what about PER for any loss, distributional RL or twin critics ?
- Supervised Learning: importance sampling to reduce variance of the stochastic gradient
- This paper:
  - PER = importance sampling in an heuristic way
  - This heuristic can be improved: Large Batch Experience Replay

# RL basics

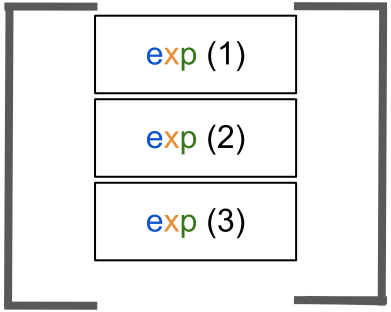
DQN:

- Replay buffer of size $N$
- Neural network $Q_\theta$
- Target network $Q_{targ}$
- Uniform sampling for SGD step on the loss:

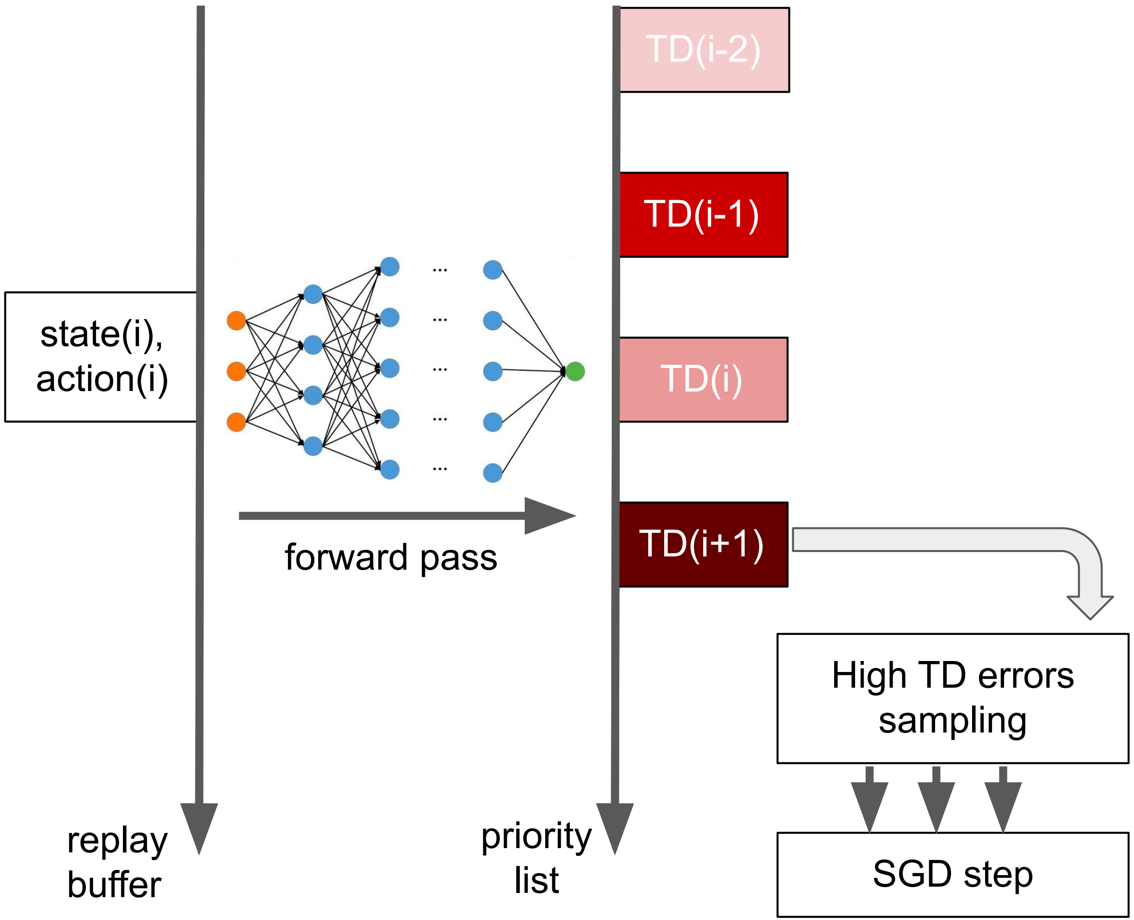
$$\frac{1}{N}\sum_{i=1}^{N}\left(r_i + \gamma \max_{a'} Q_{targ}(s'_i, a') - Q_\theta(s_i, a_i)\right)^2$$




Replay buffer



# **Prioritized Experience Replay** (Schaul et al., 2016)




PER samples transitions with high TD error

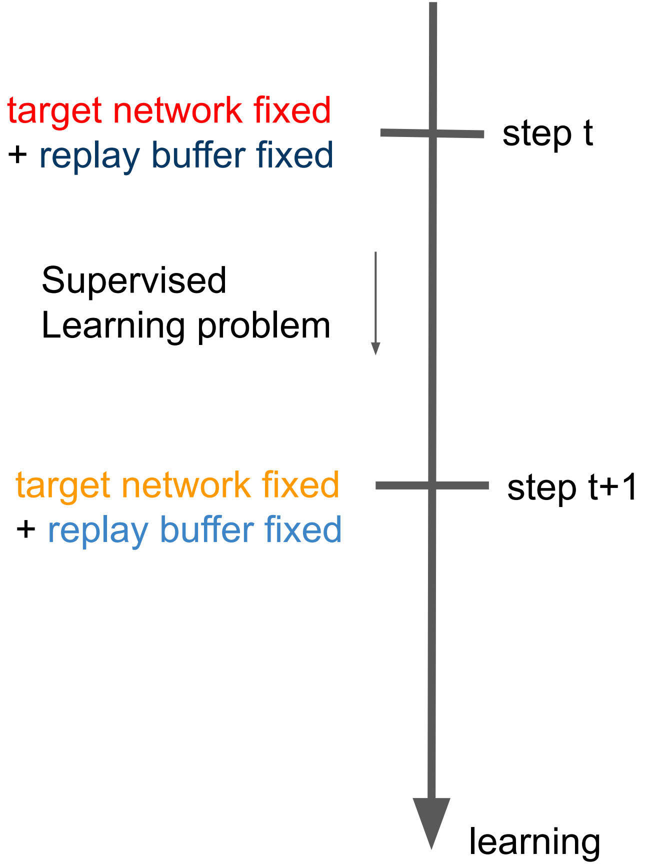
$$p_i^{PER} \propto |Q_\theta(x_i) - y_i|$$

$$\begin{cases} x_i & = (s_i, a_i) \\ y_i & = r_i + \gamma \max_{a'} Q_{targ}(s'_i, a') \end{cases}$$

What about PER for:

- Any loss?
- Distributional RL?
  - Rainbow (Hessel et al., 2018) → loss as priority
- Twin critics → Two TD errors…

# Importance Sampling for Approximate Dynamic Programming




In supervised learning, equivalence between:

High convergence speed

⟺

Low variance of the gradient estimate

⟺

Appropriate sampling distribution

# Importance Sampling for Supervised Learning

- Let $p$ be the sampling scheme and $G_i$ the per-sample gradient
- Minimize the variance of the stochastic gradient by solving: $\min_p \mathbb{E}_{i \sim p}[G_i^\top G_i]$
- Optimal sampling scheme is: $p_i^* \propto \|\nabla_\theta \ell(Q_\theta(x_i), y_i)\|_2$
- Highest convergence speed!
- BUT:
  - Computation requires forward AND backward pass
  - Must be done for ALL collected samples

$\rightarrow$ **Impractical**

# PER: two approximations

- Approximate priorities

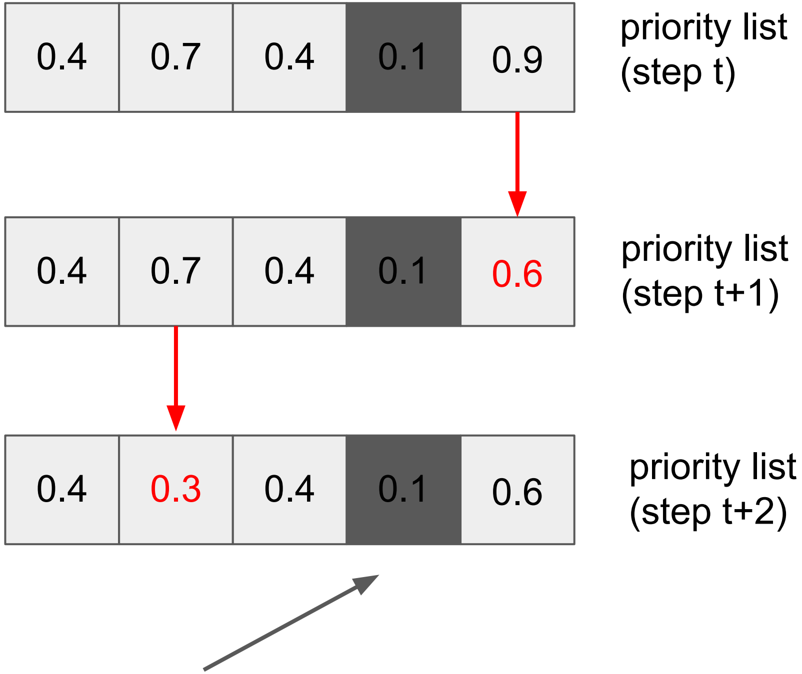
- If L2 loss, $p_i^* \propto \|\nabla_\theta \ell(Q_\theta(x_i), y_i)\|_2$

$$= \underbrace{|Q_\theta(x_i) - y_i|}_{\text{TD error}} \times \|\nabla_\theta Q_\theta(x_i)\|_2$$

- If $\|\nabla_\theta Q_\theta(x_i)\|_2$ is constant, then

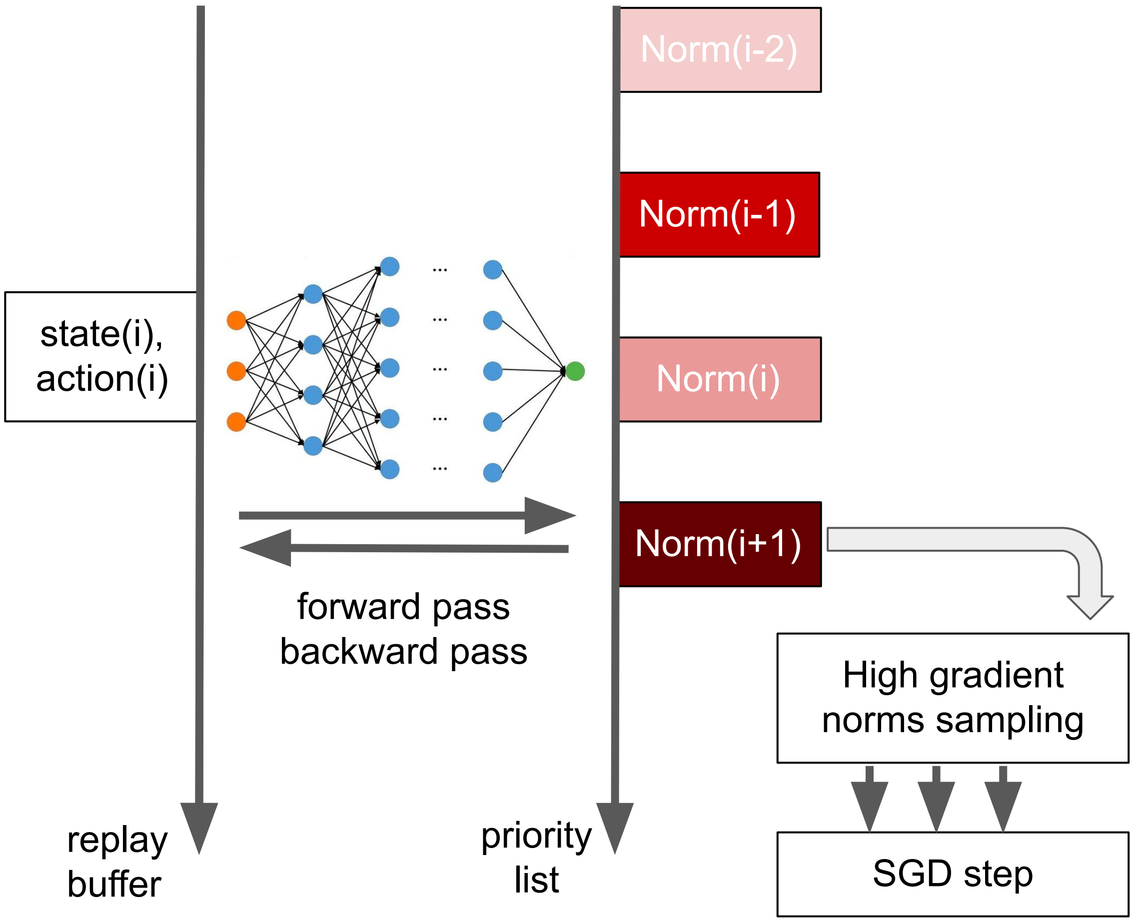
$$p_i^* \propto \|\nabla_\theta \ell(Q_\theta(x_i), y_i)\|_2 \propto |Q_\theta(x_i) - y_i|$$

- Outdated priorities




low priority might never change…
and associated sample never selected

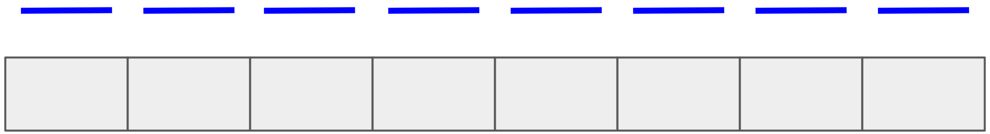
# Gradient Experience Replay



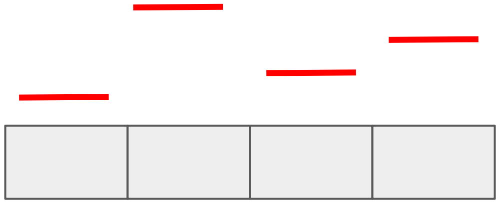

- Same data structure as PER
- Identical hyper-parameters
- Uses per-sample gradient norms instead of TD errors

# Large Batch Experience Replay

samples uniformly replay buffer



downsamples with per-sample gradient norms



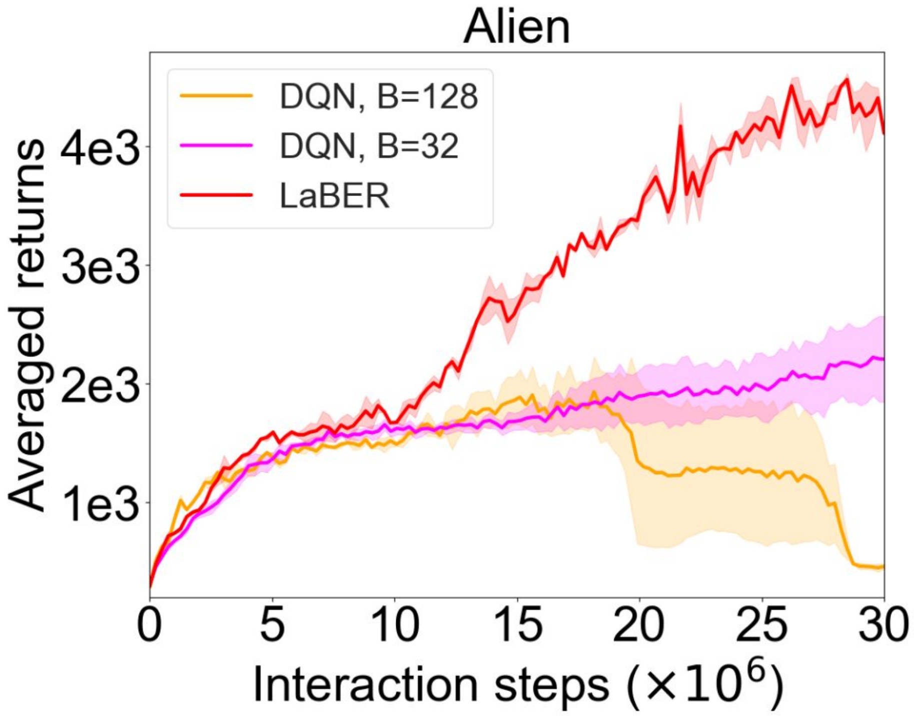
SGD step

**LaBER algorithm**

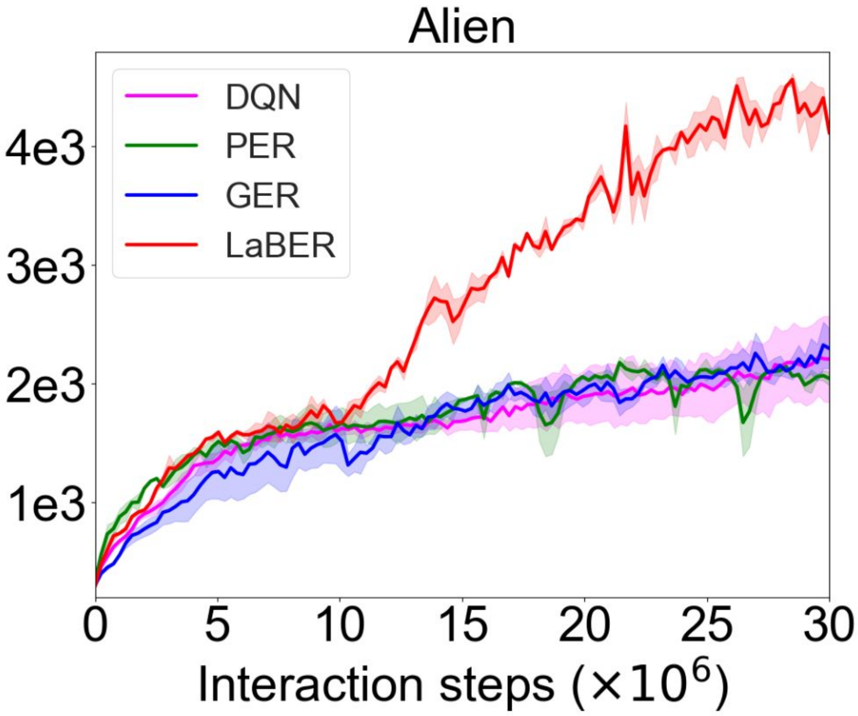
1. Sample uniformly a large batch
2. Compute exact per-sample gradient norms
3. Downsample to a mini-batch according to per-sample gradient norms computed
4. Perform SGD step on mini-batch

# Experimental results

- What about just a larger mini-batch ?

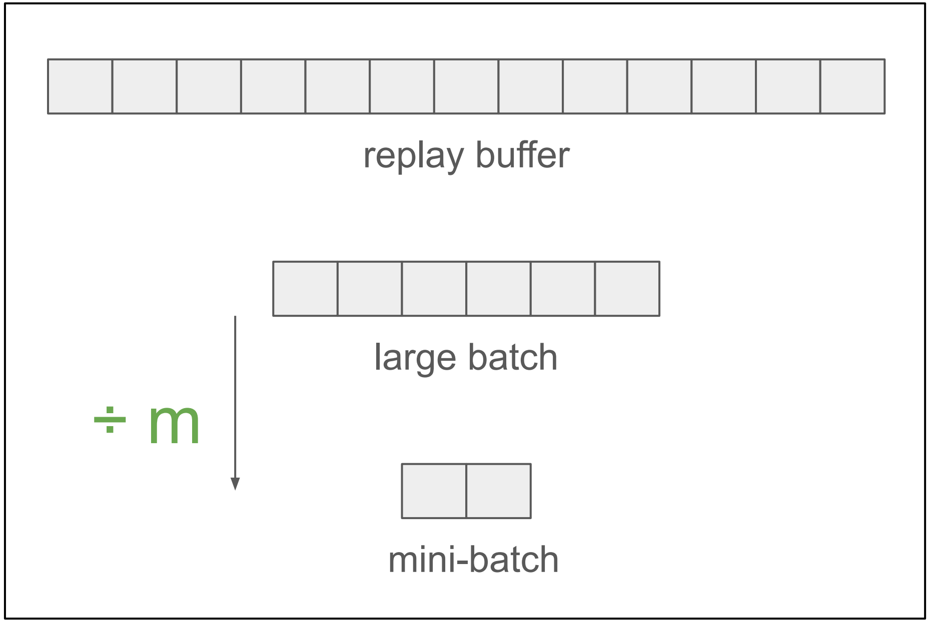



- PER vs LaBER ?



# Large Batch Experience Replay: key advantages

- Just one hyperparameter: the factor m between the large batch and the mini-batch
- Improvements over vanilla DQN or PER
- Easy to code and backed by theory
- Straight extension to:
    - any loss function
    - distributional RL
    - twin critics



# Conclusion

- PER: heuristic performing variance reduction of the stochastic gradient
- LaBER yields improvement with:
  - less hyperparameters
  - less code to write
  - broad application possibilities

**More information in the paper!**