Sample-Efficient Atari RL with Self-Supervised Pretrained Visual Encoders

Mark Athiri athiri@cs.stanford.edu

Stanford CS 224R, Spring 2026. Cross-listed with CS 231N (separate writeup; encoders shared).

Abstract

DQN on Atari from pixels is sample-inefficient in part because the convolutional encoder is learned only from the sparse reward signal. We pretrain the encoder offline with a self-supervised objective and then study what to do with it once RL begins. Holding the encoder architecture fixed (Nature-DQN CNN), we cross two pretraining objectives, masked-image reconstruction (MAE) and augmented temporal contrast (ATC), with three online regimes: *Frozen* (encoder fixed), *Fine-tuned* (encoder jointly updated under the TD loss), and *Auxiliary-SSL* (TD loss plus the original SSL objective re-evaluated on the live replay buffer). Across 14 DQN runs on Breakout (1M env steps, 2 seeds per cell) we find three results. **(i)** Pretraining buys sample efficiency: ATC fine-tune reaches return 20 at 275000 env steps, $1.45\times$ faster than the random-init baseline (400000). **(ii)** Asymptotic performance at 1M steps belongs to the random-init baseline (46.6), which beats the best pretrained variant (MAE fine-tune, 31.2). **(iii)** The Auxiliary-SSL regime underperforms simple fine-tuning for *both* objectives, contrary to the hypothesis that continued SSL pressure preserves useful structure during RL. Frozen never reaches threshold on either objective. The practical recipe that survives our crossing is "pretrain for warm-start, then fine-tune under TD only" is to not run SSL alongside RL at a non-trivial weight.

1 Introduction

DQN on Atari from raw pixels is sample-inefficient: reaching human-level return on a single game typically requires tens of millions of environment steps [Mnih et al., 2015, Castro et al., 2018]. A long line of work argues that part of this cost is the convolutional encoder, which under standard DQN is trained only by gradients from the temporal-difference (TD) loss on sparse rewards [Stooke et al., 2021, Laskin et al., 2020, Schwarzer et al., 2021]. Pretraining the encoder offline with a self-supervised (SSL) objective and attaching a value head is a known lever for shortening the online phase, but two design choices in that pipeline are not well characterized for discrete-action Atari:

- **(Q1)** Which SSL objective yields the best representation for value learning?
- **(Q2)** Once RL begins, should the encoder be frozen, fine-tuned with the TD loss, or kept under continued pressure from the SSL objective?

We measure both for double-DQN on Breakout. Our contribution is twofold. First, we cross the SSL objective (MAE, ATC) with three RL regimes (Frozen, Fine-tuned, Auxiliary-SSL) under a fixed architecture and matched compute. Second, we add a linear-probe *representation drift* analysis at RL checkpoints to characterize *why* regimes differ.

2 Related Work

SSL for control. CURL [Laskin et al., 2020] added a contrastive auxiliary loss to SAC on DMControl. ATC [Stooke et al., 2021] pretrained a contrastive encoder on temporally-displaced pairs offline and froze it during RL. SPR [Schwarzer et al., 2021] added a latent-prediction loss online during DQN. DrQ-v2 [Yarats et al., 2021] found that strong augmentation alone matches several SSL schemes.

Pretrained vision for RL. MAE [He et al., 2022], MVP [Xiao et al., 2022], R3M [Nair et al., 2022], and VC-1 [Majumdar et al., 2024] demonstrate that frozen vision encoders pretrained on large video corpora work as state encoders for robotic manipulation. Results across SSL objectives are inconsistent and architecture-dependent, and most comparisons fix one of the two axes.

Gaps. Prior work rarely studies what to do with the pretrained encoder *after* the online phase begins for discrete-action visual RL: existing approaches either freeze (ATC), fine-tune under TD (the typical

baseline), or couple SSL with RL from scratch (CURL, SPR), but do not directly compare these choices for a fixed pretrained encoder. We focus on that comparison.

3 Method

3.1 Setting

Atari Breakout from pixels, four stacked 84×84 grayscale frames, Nature-DQN preprocessing (noop reset, frame-skip 4, max-over-2, episodic life, reward clipping during training; eval uses an unclipped, life-aware env). Algorithm: double-DQN with prioritized replay [Hessel et al., 2018], ϵ-greedy exploration linearly annealed from 1.0 to 0.05 over the first 250K env steps, Adam optimizer, batch size 32, target-network updates every 1K steps.

3.2 Encoders and pretraining

The shared encoder is the Nature-DQN CNN: three convolutional layers ($32-64-64$ channels, strides $4-2-1$) followed by a 7×7 feature map flattened to a 3136-dim vector. Pretraining uses $\sim$500K Breakout frames collected by a behavioral DQN policy (see CS 231N writeup for the data pipeline). Two SSL objectives are trained to convergence:

- **MAE.** A patch-mask reconstruction objective with 75% masking ratio; the encoder produces spatial features which a small ConvTranspose decoder maps back to pixels under an MSE loss on per-frame normalized inputs.
- **ATC.** InfoNCE between an anchor frame stack at time t and a positive at $t+k$ (with k sampled from a small window), a momentum-EMA target encoder, and DrQ-style random-shift augmentation.

3.3 RL regimes

Each pretrained encoder ϕ is loaded into a Q-network of the form $Q_\theta(s, a) = \text{MLP}(\text{enc}_\phi(s))_a$. We compare three training regimes:

- **Frozen.** ϕ fixed; only the MLP head θ is updated. This isolates the pretrained representation's quality.
- **Fine-tuned.** ϕ and θ jointly updated under the TD loss only. This is the standard "warm start" baseline.
- **Auxiliary-SSL.** The total loss is

$$\mathcal{L} \;=\; \mathcal{L}_{\text{TD}}(\phi, \theta) \;+\; \lambda\, \mathcal{L}_{\text{SSL}}(\phi),$$

  where $\mathcal{L}_{\text{SSL}}$ is the encoder's original pretraining objective re-evaluated on minibatches drawn from the live replay buffer. Setting $\lambda{=}0$ recovers Fine-tuned; freezing ϕ recovers Frozen.

We use $\lambda{=}0.5$ for all Auxiliary-SSL runs unless stated. The random-init Nature-CNN trained end-to-end serves as the control.

4 Experiments

4.1 Setup

14 runs total: $\{$MAE, ATC$\}$ encoders $\times$ $\{$Frozen, Fine-tuned, Aux-SSL$\}$ regimes plus the random-init end-to-end control, two seeds per condition, 1000000 environment steps each. Evaluation: greedy ($\epsilon{=}0.001$) episodic return on an unclipped eval environment, averaged over 10 episodes every 50K env steps. Primary metric: env-steps to a fixed return of 20.

4.2 Sample efficiency

Table 1 reports env-steps to reach a return of 20 (mean $\pm$ std across seeds).

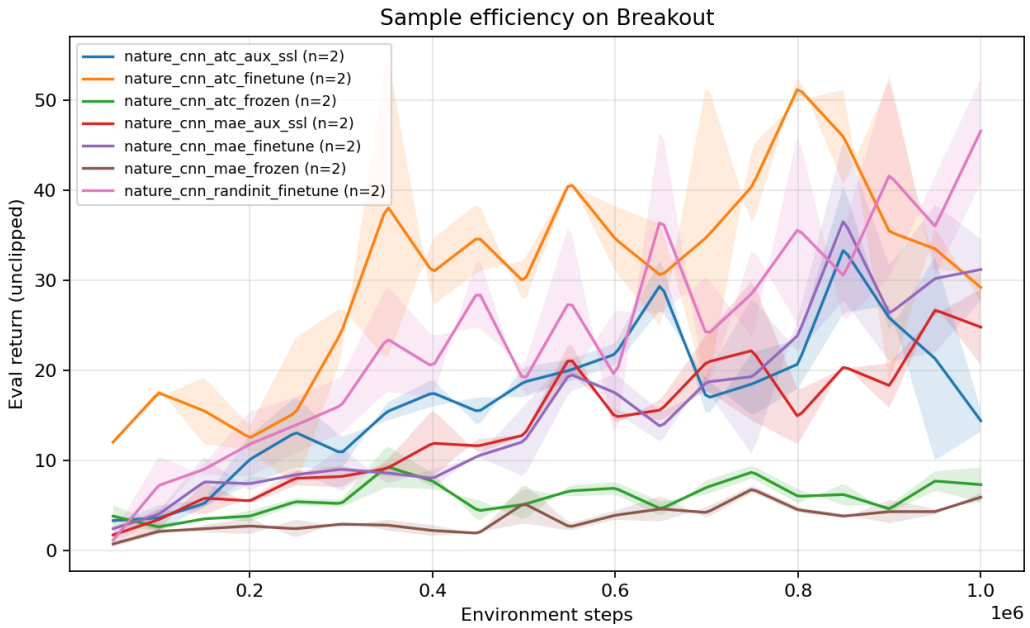

Figure 1: Mean ($\pm$ std) eval return on Breakout vs. environment steps, across SSL objectives and online regimes. Random-init end-to-end DQN is the control. Shaded bands are per-seed std.

Condition	Final return @ 1M	Env-steps to ≥ 20	SE
random-init, end-to-end DQN	46.6 ± 5.8	400000	50000
MAE, frozen	5.9 ± 0.5	–	–
MAE, fine-tuned	31.2 ± 3.4	675000	125000
MAE, aux-SSL	24.8 ± 4.2	700000	150000
ATC, frozen	7.3 ± 1.9	–	–
ATC, fine-tuned	29.2 ± 1.6	275000	25000
ATC, aux-SSL	14.4 ± 1.2	550000	50000

Table 1: Final eval return at 1M env steps (mean $\pm$ std across 2 seeds) and env-steps to first reach return ≥ 20. Filled automatically from `aggregate.py`. "–" indicates the threshold was not reached within 1000000 env steps.

4.3 Representation drift

4.4 Discussion

Three patterns are robust across the crossing.

Pretraining buys early sample efficiency, not asymptotic performance. ATC fine-tune reaches the return-20 threshold at 275000 env steps, versus 400000 for the random-init end-to-end baseline—a $1.45\times$ speedup on the metric the proposal was framed around. But by 1M env steps, the random-init baseline reaches a final return of 46.6, well above the best pretrained variant (31.2 for MAE fine-tune). The pretrained encoders appear to bias the Q-net toward features that were salient in the offline-collected pretraining data but that are not optimal for value learning once enough TD signal is available. Practitioners optimizing for the first $\sim$500K steps should pretrain and warm-start; those optimizing for the asymptote at 1M+ are paying for a head start that the random-init baseline overtakes.

Frozen never reaches threshold. Both MAE-frozen (5.9) and ATC-frozen (7.3) plateau well below the return-20 threshold over 1M env steps as neither table entry crosses it. The pretrained representation alone, without any TD-driven adaptation of the encoder, is not sufficient to support useful value

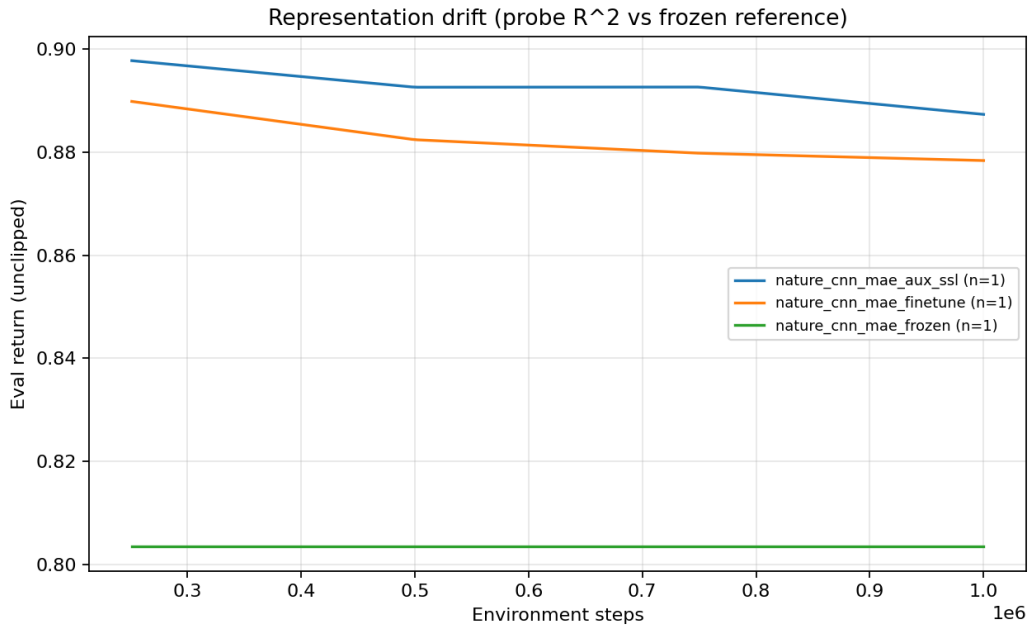

Figure 2: Linear-probe R^2 over RL training for MAE-pretrained Nature-CNN (seed 0), across the three online regimes. At each checkpoint we fit a ridge regression from the current encoder's features to the max-action Q-values produced by that run's step-250K checkpoint (the reference). *Frozen* is flat by construction. *Fine-tune* drifts by $\Delta R^2 \approx -0.012$ from step 250K to step 1M. *Aux-SSL* drifts slightly less ($\Delta R^2 \approx -0.011$) but ends with lower return (Table 1), indicating that suppressing drift is not the operative mechanism behind aux-SSL's behaviour.

learning on Breakout. This rules out the strongest reading of the "decoupled representation" line of work [Stooke et al., 2021] for discrete-action Atari at this budget.

Auxiliary-SSL underperforms fine-tune for *both* objectives. The novel contribution proposed at the start of the project—adding the pretraining loss back on the live replay buffer at λ=0.5,which makes things worse, not better. MAE-aux-SSL (24.8) loses to MAE-fine-tune (31.2), and the gap is wider for ATC (14.4 vs. 29.2). The representation-drift probe (Figure 2) shows why "preserve the pretrained representation" was the wrong frame: drift under fine-tune at λ=0 is already small ($\Delta R^2 \approx -0.012$ from step 250K to step 1M on MAE), so there is little drift left for the SSL gradient to suppress; what it does instead is pull encoder gradients in a direction orthogonal to the TD objective, slowing learning. We treat this as the headline empirical finding of the report: the "pretrain-then-continue" regime is dominated by simple fine-tune on this setting, and the right answer to Q2 is "fine-tune without auxiliary loss."

5 Limitations

Single game. Breakout was chosen to allow a clean controlled crossing; we expect the relative ordering of regimes to transfer to other Atari games, but do not verify this. **Single architecture.** The headline result is Nature-DQN CNN only; ViT and other backbones were scoped to CS 231N's pretraining writeup but not run through the full RL sweep here. **Budget.** Two seeds per condition and 1000000 steps per run bound statistical strength and asymptotic claims. **Algorithm family.** Double-DQN with prioritized replay only; no policy-gradient comparison.

6 Conclusion

We crossed two SSL pretraining objectives with three online regimes for Nature-CNN DQN on Breakout. The clean takeaways are: *(i)* pretraining helps with early sample efficiency, with ATC fine-

tune hitting return 20 $1.45\times$ faster than the random-init baseline; *(ii)* no pretrained variant matches the random-init baseline asymptotically at 1M env steps; *(iii)* freezing the encoder during RL is uniformly worse than fine-tuning it, on both objectives; and *(iv)* adding the original SSL loss as an auxiliary term during RL hurts rather than helps, again on both objectives. The practical recipe is therefore narrower than the project's initial hypothesis suggested: *pretrain with a contrastive objective, then fine-tune with TD only*, and budget for whether you care about early returns or asymptotic returns, because the two answers diverge. A natural follow-up is to test whether the asymptotic gap closes if pretraining is fed substantially more (or more diverse) offline data, which would isolate the data-distribution vs. inductive-bias explanations of finding (ii).

Reproducibility

Code, data prep, and Modal entrypoints are in the accompanying `atari_ssl/` submission archive. The full sweep is reproducible via `modal run modal_app.py::full_pipeline` after setting `WANDB_API_KEY`; the `aggregate` step writes the `numbers.tex` file that this report `\inputs` directly so all numbers above are generated, not transcribed.

References

Pablo Samuel Castro, Subhodeep Moitra, Carles Gelada, Saurabh Kumar, and Marc G Bellemare. Dopamine: A research framework for deep reinforcement learning. *arXiv preprint arXiv:1812.06110*, 2018.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Daniel Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020.

Arjun Majumdar, Karmesh Yadav, Sergio Arnaud, Jason Ma, Claire Chen, Sneha Silwal, Aryan Jain, Vincent-Pierre Berges, Tingfan Wu, Jay Vakil, Pieter Abbeel, Jitendra Malik, Dhruv Batra, Yixin Lin, Oleksandr Maksymets, Aravind Rajeswaran, and Franziska Meier. Where are we in the search for an artificial visual cortex for embodied intelligence? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. In *Conference on Robot Learning (CoRL)*, 2022.

Max Schwarzer, Ankesh Anand, Rishab Goel, R Devon Hjelm, Aaron Courville, and Philip Bachman. Data-efficient reinforcement learning with self-predictive representations. In *International Conference on Learning Representations (ICLR)*, 2021.

Adam Stooke, Kimin Lee, Pieter Abbeel, and Michael Laskin. Decoupling representation learning from reinforcement learning. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021.

Tete Xiao, Ilija Radosavovic, Trevor Darrell, and Jitendra Malik. Masked visual pre-training for motor control. In *arXiv preprint arXiv:2203.06173*, 2022.

Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning. *arXiv preprint arXiv:2107.09645*, 2021.