Transformer-Assisted Representation Learning for Deep Reinforcement Learning

Unique Divine (u.divine@columbia.edu),
Erik Skalnes (eas2301@columbia.edu)

Columbia University

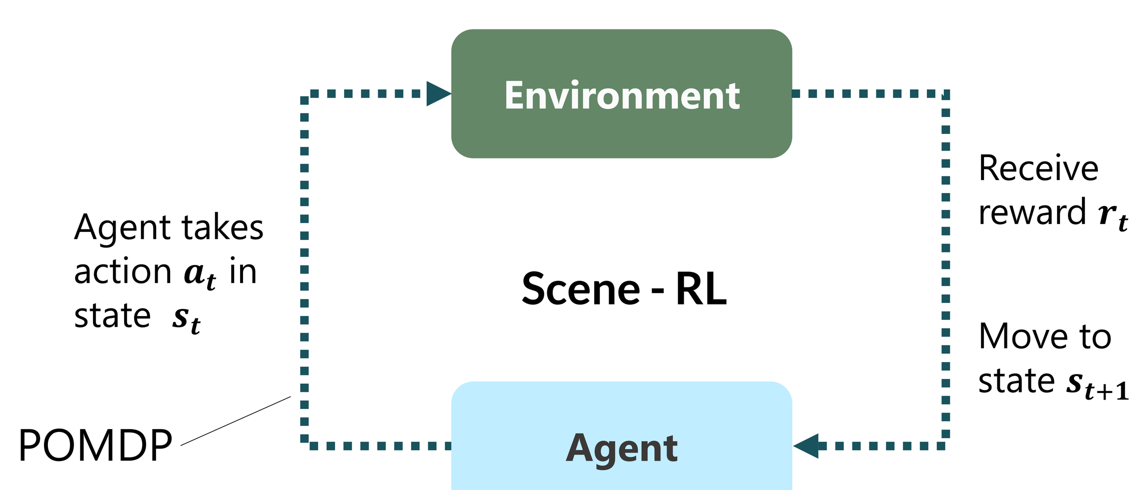
Motivation

- Raw sensory inputs often have high dimensionality, making them difficult to work with in the context of reinforcement learning (RL):
  - Sample-inefficient
  - Computationally heavy
- Auxiliary tasks have been shown to enhance state representations in RL → better performance

Objectives

- Improve sample efficiency in sparse-reward, partially observable environments
- Train for an unsupervised auxiliary task to efficiently learn better latent representations for image inputs
- Perform unsupervised learning and reinforcement learning simultaneously.

RL (Background)



After each scene in an episode, append the tuple (s_t, a_t, r_t, s_{t+1}) to the experience replay buffer, $\mathcal{B}$.

Self-Supervised Pre-training (Background)

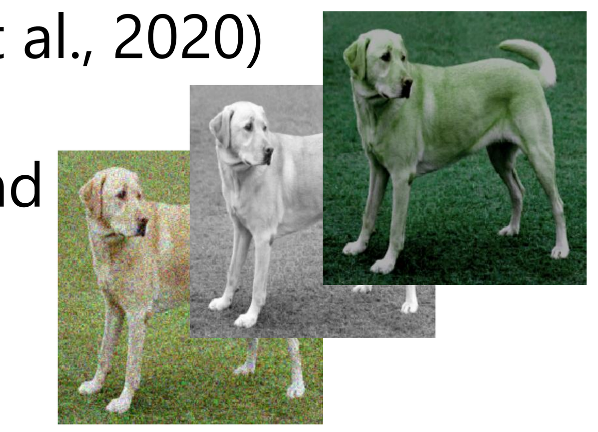
(BERT) Bidirectional Encoder Representations for Transformers

- Trained with "masked language model" objective: **Mask some percentage of the input** sequence and **predict the masked elements**.

"You only [mask] once." →  → "live"

Contrastive learning:

- Self-supervised pre-training method
- Effective in computer vision for training image encoders
  - Encoder: A map from high-dimensional inputs to low-dimensional representations
- Allows models with pure pixel inputs to nearly match SOTA performance and sample-efficiency in deep RL (Srinivas et al., 2020)
- Goal: **Learn a representation space** in which different representations of the **same image** are **close together** and representations of **different images** are **far apart**.



Methods

Masked Contrastive Learning

Step 1. **Sample S** $= (s_1, \ldots, s_T)$ from the replay buffer $\mathcal{B}$.

Step 2. **Mask S. → get S'** $= (s'_1, \ldots, s'_T)$, where $s'_i = \bar{s}'_i M_i + s_i(1 - M_i)$, $M_i \in \{0,1\}$, and $\mathrm{p_M} \coloneqq \mathbb{P}(M_i = 1)$.

Step 3. **Encode S'**.
$f_\theta(S') = (f_\theta(s'_1), \ldots, f_\theta(s'_T))$

Step 4. **Reconstruct masked inputs** using the context from the global input.

$f_\theta(S')$ → Transformer encoder φ

Architecture:
- Blocks of self-attention followed by fully-connected layer(s)
- Residual connections
- Layer normalization

Step 5. **Train encoders f_θ and φ** using contrastive loss.

$$\mathcal{L}_{ct} = \sum_{i=1}^{T} -M_i \log\left(\frac{\exp(q_i \cdot k_i/\tau)}{\sum_{j=1}^{T} \exp(q_i \cdot k_j/\tau)}\right)$$

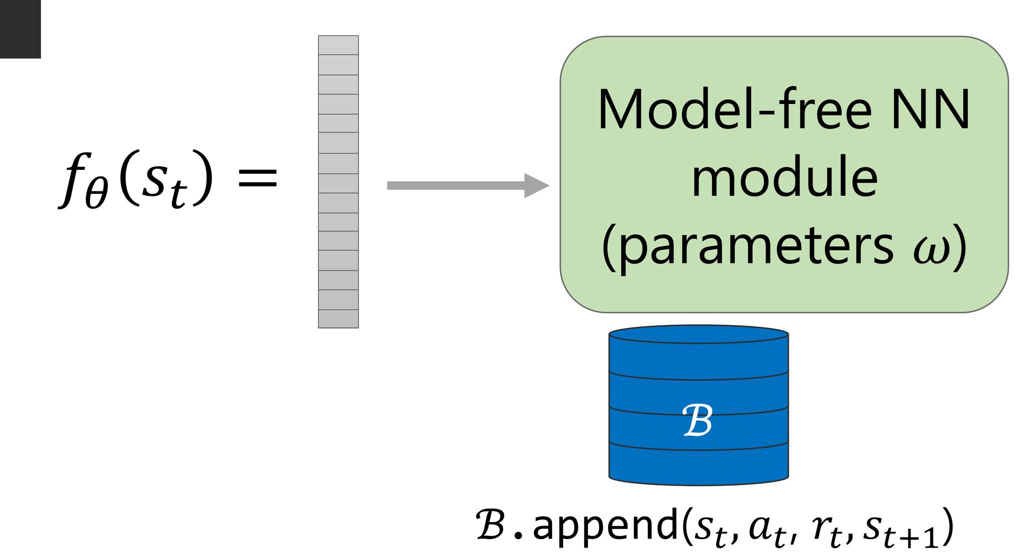
Queries $q_i = \varphi(f_\theta(s'_i))$
Keys $\mathrm{k_i} = f_{\theta_k}(s_i)$

Update θ_k with momentum contrast (He et al., 2019): $\theta_k = m\mathrm{SG}(\theta) + (1-m)\theta_k$

Overall objective: $\min_{\theta,\omega,\varphi} \mathcal{L} = \mathcal{L}_{RL} + \lambda\mathcal{L}_{ct}$

Reinforcement Learning

- Select and take actions
- Populate the replay buffer with experiences
- Train ω with objective $\mathcal{L}_{RL}$



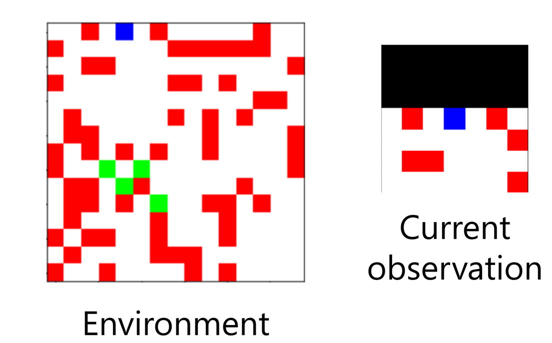
Methods Key

f_θ (ConvNet): The RL encoder. Also, the query encoder for contrastive learning
f_{θ_k} (ConvNet): The key encoder for contrastive learning. Uses the same architecture as f_θ but has different parameters
λ, τ, K, p_M : Hyperparameters
φ (Transformer): Encodes masked inputs sequences
ω (NN module): Parameters for the learnable policy or action value functions. Used in $\mathcal{L}_{RL}$.

References

- Chen et al. (2020). A simple framework for contrastive learning of visual representations. *In International conference on machine learning* (pp. 1597-1607). PMLR. [pdf]
- Devlin et al. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*. [pdf]
- Hessel et al. (2018). Rainbow: Combining improvements in deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 32, No. 1).
- Mnih et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540), 529-533. [pdf]
- Srinivas et al. (2020). CURL: Contrastive unsupervised representations for reinforcement learning. In *International Conference on Machine Learning* (pp. 5639-5650). PMLR. [pdf] [code]
- Oh et al. (2016). Control of memory, active perception, and action in Minecraft. In *International Conference on Machine Learning* (pp. 2790-2799). PMLR. [pdf]
- Oord et al. (2018). Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*. [pdf]
- Zhu et al. (2020). Masked Contrastive Representation Learning for Reinforcement Learning. *arXiv preprint arXiv:2010.07470*. [pdf] [code]

Toy environment:



Ongoing & Future Experiments

- Benchmark on more challenging environments after tinkering on the toy environment.
- Use the transformer to prioritize sequences on the buffer.
- Diversify the replay buffer and see if that improves training.

Open-source code coming soon to https://github.com/eskalnes/RL_memory