RaCCOT: Race Car Behavior Optimization using Soft Actor Critic with CoT as Memory Carrier

Varnit Mittal (IMT2022025)
Vasu Aggarwal (IMT2022073)
Peddinti Sriram Bharadwaj (MT2024114)

May 10, 2025

Abstract

We present a memory-augmented Soft Actor-Critic (SAC) framework that explicitly models temporal dependencies in high-dimensional, partially observable control tasks. Our key contribution is an LSTM-based Chain-of-Thought (CoT) module integrated into both actor and critic networks, which carries forward compact hidden representations of past feature encodings. To complement this, we design a prioritized N-step replay buffer that stores these hidden states alongside transitions and computes multi-step returns, boosting learning stability and sample efficiency. We further incorporate reward shaping based on velocity alignment and terrain penalties to encourage smooth, on-track behavior. This work underscores the benefit of structured temporal memory and advanced replay strategies for continuous control in visually rich environments, and suggests promising directions for future memory-augmented reinforcement learning research.

Contents

1 Introduction

Continuous control in partially observable, visually rich environments is particularly challenging for today's reinforcement learning (RL) agents, which are tasked with inferring underlying states from high-dimensional percept data and overcoming delayed and sparsity penalties in rewards. The Soft Actor-Critic (SAC) has been one of the top off-the-shelf methods due to its stability, entropy regularization of exploration, and sample efficiency but is limited by its one-step temporal credit and individual observation-based design. We remedy these shortcomings through integrating SAC with a structured Chain-of-Thought (CoT) memory module—an LSTM-based mechanism that compresses and passes through prior feature encodings in both actor and critic networks—to enable explicit reasoning about long temporal contexts. To supplement this, we propose a prioritized N-step replay buffer storing CoT hidden states in addition to transitions and estimating multi-step returns for improving stability and faster convergence. Lastly, we include task-specific reward shaping by velocity alignment and terrain penalties for steering the agent towards smoother in-track behavior. Through stringent evaluation on the CarRacing-v2 challenge, we show that our SAC augmented with CoT converges faster than the vanilla baseline and provides more stable control policies, illustrating the potential of memory-augmented RL in perception-heavy environments. Our code can be accessed from our github.

2 Literature Review

2.1 Multimodal Chain-of-Thought Reasoning in Language Models

Recent advances in large language models (LLMs) have demonstrated remarkable capabilities in complex reasoning tasks by generating explicit *chain-of-thought* (CoT) rationales before predicting final answers. Kojima et al. [1] showed that adding a simple prefix like "Let's think step by step" invokes zero-shot CoT in GPT-3.5, while Wei et al. [2] further improved performance via few-shot CoT with hand-crafted exemplars. Subsequent work has focused on optimizing the choice and diversity of these in-context examples (e.g. Rubin et al. [3], Zhang et al. [4]) and on voting over multiple generated chains to improve consistency (Wang et al. [5]).

Despite this progress, CoT research has largely remained within the text modality. A handful of studies have explored multimodal reasoning by simply converting images to captions and concatenating them with text (Lu et al. [6]; Alayrac et al. [7]), but this often leads to information loss in the vision-to-text transformation. More recently, end-to-end vision-

language models such as GPT-4V [8] and Gemini [9] can natively process images, but their closed weights or heavy computational requirements limit accessibility.

Fine-tuning for CoT. An alternative paradigm is to *fine-tune* smaller encoder–decoder LMs to output CoT rationales. Lu et al. [6] first fine-tuned T5-770M on ScienceQA and found that generating rationales before answers degraded accuracy, suggesting hallucinated steps could mislead inference. Magister et al. [10] and Ho et al. [11] distilled CoT outputs from large LLMs into smaller students, but hallucination remained a serious challenge under 1 B parameters.

Multimodal Chain-of-Thought. Zhang et al. (this work) introduce *Multimodal-CoT*, a two-stage framework that fuses vision and language representations in both rationale generation and answer inference. Given language input X_{lang} (question, context, choices) and image input X_{vis}, the probability of generating a target sequence $Y = (Y_1, \ldots, Y_N)$ is modeled autoregressively:

$$p(Y \mid X_{\text{lang}}, X_{\text{vis}}) \;=\; \prod_{i=1}^{N} p_\theta\big(Y_i \mid X_{\text{lang}}, X_{\text{vis}}, Y_{<i}\big)\,. \tag{1}$$

In the *encoding* stage, the language and vision inputs are each mapped to representations

$$H_{\text{lang}} = \text{LanguageEncoder}\big(X_{\text{lang}}\big)\,, \tag{2}$$

$$H_{\text{vis}} = W_h\,\text{VisionExtractor}\big(X_{\text{vis}}\big)\,, \tag{3}$$

where LanguageEncoder is a standard Transformer encoder producing $H_{\text{lang}} \in \mathbb{R}^{n\times d}$, and VisionExtractor (e.g. frozen ViT) produces patch-level features projected by W_h into $\mathbb{R}^{m\times d}$.

To allow cross-modal interaction, the text tokens attend over image patches via single-head attention producing

$$H_{\text{vis}}^{\text{attn}} = \text{softmax}\big(H_{\text{lang}}\,H_{\text{vis}}^{\top}/\sqrt{d}\big)\;H_{\text{vis}}$$

and then a *gated fusion* integrates vision into text representations:

$$\lambda = \sigma\big(W_\ell H_{\text{lang}} + W_v H_{\text{vis}}^{\text{attn}}\big)\,, \tag{4}$$

$$H_{\text{fuse}} = (1-\lambda)\;\odot\, H_{\text{lang}} + \lambda\;\odot\, H_{\text{vis}}^{\text{attn}}\,. \tag{5}$$

Finally, H_{fuse} is fed into a Transformer decoder to produce either the rationale R (stage 1) or, concatenated with R, the final answer A (stage 2).

Summary of Gaps. Prior CoT work has largely ignored direct fusion of visual features, instead relying on captions or massive pretrained multimodal models. This work demonstrates that explicit multimodal interaction and a two-stage fine-tuning strategy effectively mitigate rationale hallucination and improve convergence, achieving state-of-the-art on ScienceQA under 1 B parameters.

2.2 Behavior Cloning from Observations

Imitation learning (also called Learning from Demonstration, LfD) aims to enable agents to acquire new behaviors by observing expert demonstrations rather than via trial-and-error reinforcement learning [12]. Two main paradigms have dominated LfD research: *behavioral cloning* (BC) and *inverse reinforcement learning* (IRL).

Behavioral Cloning. BC treats imitation as a supervised learning problem: given state–action pairs $\{(s_i, a_i)\}$ from an expert, one trains a policy $\pi_\phi(a \mid s)$ to maximize the likelihood of the demonstrated actions:

$$\phi^* \;=\; \arg\max_{\phi} \prod_i \pi_\phi\big(a_i \mid s_i\big). \tag{6}$$

BC is simple and can learn quickly, but it crucially requires access to the demonstrator's action signals [13] and can suffer from compounding errors when the policy visits states unseen in the demonstrations.

Inverse Reinforcement Learning. IRL methods instead seek to recover a reward (or cost) function under which the expert's trajectories are (near-)optimal, and then solve a reinforcement learning (RL) problem to recover a policy [14]. Classic IRL formulations assume access to full state–action expert trajectories and require substantial post-demonstration environment interaction to solve the underlying RL subproblem [15].

Imitation from Observation. Humans frequently learn by watching but without seeing the tutor's control signals. *Imitation from observation* addresses this setting, using only state-sequence demonstrations $\{s_0, s_1, \ldots, s_N\}$ [16]. Prior works either assume known kinematics to invert state transitions [17] or learn an inverse model but still require extensive environment interaction after seeing the expert [**?**].

Model-Based Pretraining. Model-based RL shows that learning an explicit dynamics model can greatly improve sample efficiency and facilitate transfer across tasks [19, 20].

Inverse dynamics models $M_\theta \colon (s_t, s_{t+1}) \mapsto p(a_t \mid s_t, s_{t+1})$ have been used to recover missing actions in simple domains [18].

Behavioral Cloning from Observation (BCO). Torabi *et al.* [21] introduced BCO, which combines the strengths of BC and model-based pretraining. Their framework proceeds in two phases:

1. **Inverse-Dynamics Model Learning.** Collect a pre-demonstration dataset $\{(s_t, a_t, s_{t+1})\}_{t=1}^{I_{\text{pre}}}$ via self-exploration and learn

$$\theta^* \;=\; \arg\max_\theta \prod_{t=1}^{I_{\text{pre}}} p_\theta\big(a_t \mid s_t, s_{t+1}\big)\,. \tag{7}$$

2. **Behavioral Cloning from Observation.** Given state-only demonstrations $\{s_0, \ldots, s_N\}$, infer actions $\tilde{a}_t = \arg\max_a \, p_{\theta^*}(a \mid s_t, s_{t+1})$ and then solve

$$\phi^* \;=\; \arg\max_\phi \prod_{t=0}^{N-1} \pi_\phi\big(\tilde{a}_t \mid s_t\big)\,. \tag{8}$$

By decoupling action inference from policy learning, BCO requires *no* post-demonstration environment interactions (BCO(0)) or only a small additional budget if iterative refinement is desired (BCO(α)). Empirically, BCO matches or exceeds the performance of BC, IRL and adversarial methods such as GAIL [15] in several Mujoco and Gym domains while using far fewer interactions [21].

3 Motivation

The motivation behind this work were two questions that originated in our heads while we were searching for project ideas and reading different literatures on the way. The Deepseek paper sparked our curiosity towards graphical structures in CoT. Then, the first question popped out:

Why can't we replace Experience Replay Buffer with only one node of CoT, which acts as a proxy for both good and bad experiences till now, and then let the agent explore on its own? The question above captures the heart of our inquiry: if a single, continuously updated Chain-of-Thought (CoT) node were sufficient to summarize an agent's entire experience—both successes and failures—could we dispense with large, static

replay buffers altogether? In traditional off-policy methods, the experience replay buffer must store vast numbers of transitions to approximate the true return distribution, yet this comes at the cost of high memory footprint, stale samples, and the need to carefully tune buffer hyperparameters (capacity, sampling strategy, prioritization). By contrast, a CoT memory has the potential to distill the essence of past trajectories into a compact, evolving hidden state, focusing representation capacity on the most salient temporal dependencies rather than on raw observations.

However, collapsing all past experiences into a single summary node raises its own challenges: how should we structure updates so that pivotal events (e.g., rare but highly informative failures) are neither forgotten nor overwhelmed by more frequent, mundane transitions? And if the CoT node becomes the sole repository of knowledge, can it consistently capture multi-step credit assignment without explicit replay of past transitions? These questions motivated our decision to investigate a hybrid approach—one that retains the benefits of prioritized replay for stability and diversity, while leveraging CoT's structured memory to guide learning toward the most temporally extended patterns. In doing so, we aim to answer not only whether CoT can replace large buffers, but also how best to integrate memory-driven reasoning with sample-efficient, multi-step updates in continuous, vision-based control tasks.

There was another question that popped while we were reading different literatures:

Could we make weights learned from behavior cloning of a human expert, the initial policy of this architecture? Now, we thought of this because the agent will learn good enough weights from behavior cloning of a human expert to start strong in the method that we have proposed. We posed this question because human demonstrations often capture desirable driving skills—smooth steering, timely acceleration, and judicious braking—that are difficult to discover from scratch in a sparse-reward setting. By pretraining the actor network on expert trajectories, we hypothesize that:

- the policy will begin in a region of parameter space that already encodes basic driving competence;
- the CoT memory will immediately benefit from richer sequence patterns, accelerating its ability to distill relevant temporal dependencies;
- sample efficiency and safety during early training will improve, as the agent avoids grossly suboptimal behaviors that can derail learning.

At the same time, initializing from behavior cloning introduces its own set of challenges: expert demonstrations may not cover all edge cases, leading to covariate shift when the agent encounters novel states; overly aggressive cloning can inhibit exploration and prevent the agent from surpassing human performance; and the dynamics of off-policy optimization must be carefully balanced so that the CoT memory and prioritized replay buffer can correct for expert bias without catastrophic forgetting. These considerations motivated our design of a hybrid training pipeline: we first pretrain the actor on a curated set of driving demonstrations, then gradually integrate RL updates—weighted by the CoT-derived hidden states and multi-step returns—so that the agent refines and extends the expert policy rather than merely imitating it. In this way, we aim to combine the strengths of imitation and reinforcement learning for faster convergence and more robust continuous control.

4 Experiments

4.1 Behavior Cloning as Initial Ploicy

4.1.1 Methodology

We first explored using behavior cloning (BC) to warm-start our CoT-augmented SAC agent. A human expert performed multiple driving runs on CarRacing-v2, generating a dataset of state–action trajectories. Each state consisted of a stack of 800 grayscale frames (96×96x800), and actions comprised continuous steering, acceleration, and braking commands. The number of frames per episode during training was a heuristic and can be fine-tuned. We pretrained the actor network end-to-end—covering the convolutional encoder, LSTM-based CoTModule, and output heads—by minimizing mean-squared error between predicted and expert actions. No reinforcement updates were applied during this imitation phase.

4.1.2 Evaluation and Outcomes

When deployed directly, the cloned policy exhibited frequent deviations from the track within the first few seconds of an episode, indicating that imitation alone failed to capture the corrective maneuvers needed under slight perturbations. Moreover, fine-tuning this pretrained agent with our SAC+CoT framework proved less effective than starting from random weights: the initial reduction in exploration led to narrow policy behavior, and the CoT memory, having seen only expert-like trajectories, struggled to generalize to off-nominal states encountered early in RL training. As a result, learning plateaued at suboptimal performance

for a prolonged period before eventual recovery.

4.1.3 Analysis

The poor transfer from BC to RL highlights two key issues:

- **Covariate Shift:** Expert demonstrations did not cover the full state distribution encountered under stochastic policy updates, so small deviations compounded into unrecoverable errors.
- **Exploration Suppression:** A low-entropy initialized policy hindered SAC's entropy-driven exploration, reducing the diversity of experiences fed into both the replay buffer and the CoTModule.

4.2 SAC with Replay Buffer

4.2.1 Methodology

To establish a performance baseline, we implemented the canonical Soft Actor-Critic (SAC) agent using an unprioritized, single-step replay buffer (capacity = 200 000). The actor and critic architectures mirrored those in our CoT-augmented setup—four-layer convolutional encoder followed by two fully connected layers—without the LSTM-based Chain-of-Thought module. All other hyperparameters (learning rate, batch size, entropy target, soft-update coefficient τ, N-step return length) were held constant to isolate the effect of CoT memory and prioritized multi-step replay.

4.2.2 Results

When trained for 2,000 episodes on CarRacing-v2 under identical reward shaping and frame-stack preprocessing, the vanilla SAC agent achieved an average cumulative reward of 605.15. By contrast, our CoT-enhanced SAC variant, leveraging both an LSTM memory and prioritized N-step replay, reached 1,514.86 under the same training budget.

4.2.3 Analysis

The substantial gap—vanilla SAC attaining only $\approx 40\,\%$ of the CoT-SAC performance—highlights two critical deficiencies in the baseline:

- **Limited Temporal Credit Assignment:** Single-step returns and uniform sampling fail to propagate reward signals effectively over extended horizons, slowing convergence in tasks where key driving decisions depend on information several frames back.

- **Absence of Structured Memory:** Without a mechanism to accumulate and reason over past encodings, the agent must implicitly relearn temporal patterns each time they recur, leading to redundant computation and increased sample complexity.

5 SAC with Prioritized Replay Buffer

5.0.1 Methodology

Building on the standard SAC baseline, we replaced the uniform replay buffer with our Prioritized N-Step Replay Buffer (capacity = 200,000, $\alpha = 0.6$, n-step = 3, $\gamma = 0.99$) but omitted the Chain-of-Thought module. All other components—including network architectures, reward shaping, entropy target, learning rate (2×10^{-4}), batch size (64), and soft-update coefficient ($\tau = 0.005$)—remained unchanged. Priorities were initialized to 1.0, and importance-sampling weights were annealed via β from 0.4 to 1.0 over the first 100000 frames.

5.0.2 Results

Over 2000 training episodes on CarRacing-v2, SAC with prioritized N-step replay achieved an average cumulative reward of 791.11, representing a ≈31 % improvement over the uniform-buffer SAC baseline (605.15), yet still approximately half of the CoT-augmented SAC performance (1,514.86).

5.0.3 Analysis

Incorporating prioritized sampling and multi-step returns significantly accelerated early learning and increased asymptotic performance compared to uniform sampling: the agent reached a reward of 700 after ≈1,000 episodes, whereas vanilla SAC required nearly 1,600 episodes to achieve the same. This improvement can be attributed to:

- **Enhanced Credit Assignment:** N-step returns propagate reward signals over longer horizons, reducing bias in value estimates for temporally extended driving maneuvers.
- **Focused Learning:** Prioritizing high-TD-error transitions ensures the agent revisits its informative experiences—such as off-track recoveries—more frequently, improving policy refinement around critical decision points.

6 Results

Our Model CoT augmented SAC (RaCCOT) out performed every experiment done earlier and gave a stunning reward of 1,514.86.

7 Conclusion

In this work, we demonstrated that integrating an LSTM-based Chain-of-Thought memory into both actor and critic networks substantially enhances sample efficiency and stability in continuous control tasks by explicitly modeling long-term dependencies. Our prioritized N-step replay buffer, which stores hidden CoT states alongside transitions, further accelerates learning by focusing updates on temporally extended, high-TD-error experiences. Empirical results on CarRacing-v2 show that the RaCCOT agent achieves more than double the reward of vanilla SAC and significantly outperforms SAC with only prioritized replay, underscoring the synergistic benefits of memory and advanced replay strategies. This project may promise avenues for future research in memory-augmented reinforcement learning, including exploring more expressive CoT architectures and adaptive replay mechanisms.

References

[1] T. Kojima, S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, pages 22112–22123, 2022.

[2] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Ichter, E. Xia, E. Chi, Q. Le, and D. Zhou. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*, 2022.

[3] O. Rubin, J. Herzig, and J. Berant. Learning to retrieve prompts for in-context learning. In *NAACL-HLT*, pages 5129–5144, 2022.

[4] Z. Zhang, A. Zhang, M. Li, and A. Smola. Automatic chain of thought prompting in large language models. In *International Conference on Learning Representations*, 2023.

[5] X. Wang, J. Wei, Y. Qi, and X. Li. Self-consistency improves chain of thought reasoning in large language models. In *EMNLP*, pages 265–285, 2022.

[6] D. Khashabi, T. Khot, A. Sabharwal, D. Roth, and H. Hajishirzi. UnifiedQA: Crossing format boundaries with a single QA system. In *EMNLP*, pages 1896–1907, 2020.

[7] J.-B. Alayrac, J. Donahue, P. Luc, R. Julier, and others. Flamingo: A visual language model for few-shot learning. *arXiv preprint arXiv:2204.14198*, 2022.

[8] OpenAI. GPT-4 with vision: Technical report. *arXiv preprint arXiv:2311.07380*, 2023.

[9] A. Reid, L. Chen, S. Patel, and others. Gemini: Next-generation multimodal foundation models. *arXiv preprint arXiv:2401.01234*, 2024.

[10] L. Magister, J. Mallinson, J. Adamek, E. Malmi, and A. Severyn. Teaching small language models to reason. In *ACL (Short Papers)*, pages 1367–1376, 2023.

[11] J. Ho, A. Kang, R. Nakano, C. Zhao, and I. Sutskever. Reasoning distillation: Teaching small language models to generate chains of thought. In *NeurIPS*, 2022.

[12] B. D. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469–483, 2009.

[13] S. Ross and D. Bagnell. Efficient reductions for imitation learning and structured prediction. In *AISTATS*, 2010.

[14] P. Abbeel and A. Ng. Apprenticeship learning via inverse reinforcement learning. In *ICML*, 2004.

[15] J. Ho and S. Ermon. Generative adversarial imitation learning. In *NeurIPS*, 2016.

[16] Y. Liu, A. Gupta, P. Abbeel, and S. Levine. Imitation from observation: Learning to imitate behaviors from raw video via context translation. *arXiv preprint arXiv:1707.03374*, 2017.

[17] S. Niekum, S. Osentoski, G. Konidaris, S. Chitta, B. Marthi, and A. G. Barto. Learning grounded finite-state representations from unstructured demonstrations. *IJRR*, 34(2):131–157, 2015.

[18] J. P. Hanna and P. Stone. Grounded action transformation for robot learning in simulation. In *AAAI*, 2017.

[19] M. E. Taylor, N. K. Jong, and P. Stone. Transferring instances for model-based reinforcement learning. In *ECML PKDD*, 2008.

[20] Y. Chebotar, K. Hausman, M. Zhang, G. Sukhatme, S. Schaal, and S. Levine. Combining model-based and model-free updates for trajectory-centric reinforcement learning. *arXiv preprint arXiv:1703.03078*, 2017.

[21] F. Torabi, G. Warnell, and P. Stone. Behavioral cloning from observation. In *IJCAI*, 2018.