Adaptformer: Sequence models as adaptive iterative planners

Akash Karthikeyan, Yash Vardhan Pant

University of Waterloo, Waterloo, ON, Canada
{a9karthi, yash.pant}@uwaterloo.ca

Abstract

Sequence models have emerged as an alternate paradigm for offline Reinforcement Learning (RL) with their remarkable generative capabilities. However, it struggles in cases where the trajectories only cover limited states or have sparse rewards. In scenarios with multi-task missions often involve *exploration, key pickup, room transition, and door opening*, the reward is only assigned at the end of all the tasks. We introduce Adaptformer, an adaptive planner that utilizes sequence models for sample-efficient exploration and exploitation. This framework relies on learning an energy-based heuristic, which needs to be minimized over an action sequence. It generates stochastic, goal-conditioned trajectories imposed through a lower bound on entropy, balancing the exploration and exploitation trade-off. Adaptformer aids in generalizing to unseen test scenarios via iterative re-planning through energy minimization. Empirical results over BABYAI environments demonstrate the effectiveness of Adaptformer. For example, Adaptformer outperforms the previous state-of-the-art LEAP (Chen et al. 2023) by ∼10% at BABYAI environments and adapts to long horizon tasks.[1]

Introduction

Conventional Reinforcement Learning (RL) methods, which attempt to estimate value functions, often face limitations in dealing with environments characterized by long horizons, and sparse rewards, and are susceptible to distractor signals (Hung et al. 2019), or favoring short-term goals. While sequence models address some of these shortcomings, their performance is reliant on the diversity of training data, results tend to fall short when training data is insufficient. Moreover, the sequence models do not offer a straightforward way to optimize the generated trajectories.

Drawing inspiration from the ideas in (Chen et al. 2023), where sequence models are viewed as implicit energy models, our methodology leverages these models to develop policies that adapt to previously unseen environments. This is achieved through a probabilistic objective (Zheng, Zhang, and Grover 2022), which enables a balance between exploration and exploitation. This strategy reframes the conditional generative properties of sequence models to perform iterative planning. It offers a dual advantage: it facilitates generalization on unseen environments and adaptive skill learning (i.e., obstacle unblocking). Empirical results on the adaptability of models in modified BABYAI environments validate the effectiveness of Adaptformer.

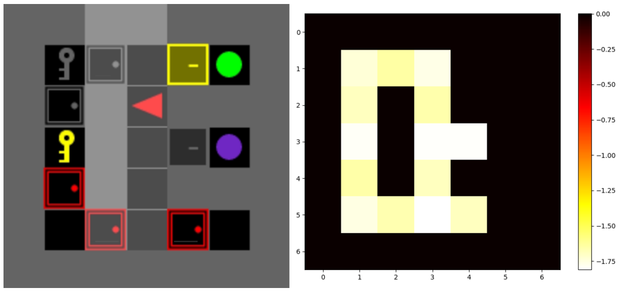

Figure 1: **Energy Landscape.** Adaptformer when conditioned with re-scaled RTG, implicitly assigns minimum energy values to sub-goals (*pick-up key, open doors*) required for task completion. The states closer to white regions (low-energy) are more likely to be transitioned into.

Related Works. Sequence modeling in RL adopts an autoregressive modeling objective. This approach leverages the conditional generative capabilities of sequence models, where conditioning on desired returns or goal states facilitates the generation of future actions leading to those states or returns. (Chen et al. 2021; Janner, Li, and Levine 2021; Parisotto et al. 2020) Following DT's (Chen et al. 2021) works on goal state conditioning, and iterative planners (Chen et al. 2023) we aim to learn a reward function that allows additional guidance and encourages novel actions. While, the probabilistic objective (Zheng, Zhang, and Grover 2022) allows the policy to adapt to unseen test environments while learning from a limited dataset.

Adaptformer: Methodology

Problem Statement. Given a trajectory $\mathcal{T}$ in the form $(\mathbf{s}_1, \mathbf{a}_1, \hat{\mathbf{R}}_1, \ldots, \mathbf{s}_n, \mathbf{a}_n, \hat{\mathbf{R}}_n)$ which comprises state, action and return-to-go (RTG) tuples, we learn the conditional probability $\pi_\theta(\mathbf{a}_t | \mathcal{T}_{\setminus t}, G)$, where, G denotes the goal states and $\mathcal{T}_{\setminus t}$ denotes the the trajectory of length $\mathcal{T}_{1:H}$ excluding the state, action and return tuple at time $\mathbf{t}$. Here H denotes the planning horizon. Our objective is to assign the demonstration trajectories minimal energy defined as the sum of negative pseudo-likelihood across the tra-

[1] Adaptformer Results

jectory $E_\theta(\mathcal{T}) = \sum_{t=1}^{H}\left[-\log \pi_\theta(\mathbf{a}_t|\mathcal{T}_{\setminus t}, G)\right]$ subject to $\mathbb{E}_{\sim\mathcal{T}}\left[\sum_{t=1}^{H}\mathcal{H}(\pi_\theta(\mathbf{a}_t|\mathcal{T}_{\setminus t}, G))\right] \geq \beta$, here $\mathcal{H}$ represents the entropy. In this manner, π_θ learns to predict actions at a given timestep using *bi-directional context* of actions across all other timesteps.

Training Objective. The problem formulation reduces to

$$\min_\theta \overbrace{\mathbb{E}_{\sim\mathcal{T}}[-\log \pi_\theta(\mathbf{a}_t|\mathcal{T}_{\setminus t}, G)]}^{L_{\text{NLL}}} - \lambda_1 \overbrace{\mathbb{E}_{\sim\mathcal{T}}[\mathcal{H}(\pi_\theta(\cdot|\mathcal{T}_{\setminus t}, G))]}^{L_{\text{CE}}}. \quad (1)$$

The Negative Log-Likelihood (L_{NLL}) corresponds to the energy term, while the L_{CE} represents the entropy component. Here, λ is a lagrangian multiplier (a.k.a. temperature), balancing exploration and exploitation. In contrast to SAC's individual transition-based approach (Haarnoja et al. 2018), Adaptformer learns the energy level over a trajectory subset with planning horizon $\mathbf{H}$ by predicting the masked action token. The average entropy across the horizon is lower bounded by β. This makes the policy less susceptible to favoring local goals and the bidirectional context-aware training avoids error accumulation. We also parameterize the action predictor network as a Gaussian distribution with diagonal covariances, where the mean and log-variance are predicted by two separate fully connected layers $\mathbf{a} \sim \mathcal{N}(\mu_\theta(\mathcal{T}_{\setminus t}), \Sigma_\theta(\mathcal{T}_{\setminus t}))$. The objective is modeled such that it allows matching the training distribution $\mathcal{T}$ while allowing some degree of mismatch which promotes exploratory actions.

Planning. Given the energy model, we estimate the energy at masked positions $\mathbf{a}_t \sim \pi_\theta(\mathbf{a}_t|\mathcal{T}_{\setminus t})$, here $\mathcal{T}_{\setminus t}$ represents the trajectories with masked actions, we update masked actions with actions corresponding to low energy value each iteration, this allows for long horizon consistent plan. The iterative planner allows for multiple sampling which seeks to minimize the total energy of the sequence. This strategy balances exploratory decisions with the exploitative nature of the energy minimization function. Adaptformer focuses on learning conditional generation of action sequences, similar to reinforcement learning via supervised learning (Emmons et al. 2022).

Results and Discussion

We use a modified BABYAI (Chevalier-Boisvert et al. 2019) environment to asses Adaptformer's generalization to test conditions: **(1) Single-goal to Multi-goal Transfer - *KeyCorridor, GoToObj*.** During training, the model learns to navigate offline trajectories towards a single goal, typically located behind a locked door. At test time, this learned capability is extended to solve multi-goal (#2) reaching problems. **(2) Auxiliary Tasks - *MiniBoss*.** The environment features additional obstacles and goal states, located across multiple rooms (#4) and looked doors.

In Table 1, we present the performance metrics that demonstrate increasing difficulty across the environments. Adaptformer outperforms the baseline model, achieving a $\sim$10%↑ due to the probabilistic objective function. Additionally, we noticed that RTG conditioning enhances the performance of LEAP, leading us to incorporate additional RTG conditioning for LEAP+RTG in our experiments. Furthermore, LEAP+RTG excels in tasks focused solely on goal-reaching ***GoToObj***, without doors or object interactions.

Env	Ours	LEAP+RTG
GoToObjMazeS4	33%	**36%**
MiniBoss	**40%**	29%
KeyCorridorS3R3	**27%**	18%

Table 1: **Comparison with Other Baselines.** The models were trained over 500 episodes and tested on 40 different environments using three distinct seeds. The model inputs consist of the agent's current position and goal positions, while LEAP additionally requires fully observable image of the environment.

Env	Ours
w/o action token + RTG	**32 %**
w/o RTG	**33 %**
w/o entropy (L_{CE})	**31 %**

Table 2: **Ablation.** Reported values correspond to mean success rate over 3 seeds. The results correspond to MiniBoss environment with an additional # 12 distractors at test time.

RTG Conditioning. We introduce a proxy reward treated as a cost-to-go, estimated with a reward network for better feature representation. This additional guidance encourages optimal planning, incentivizing novel actions like pickup, drop, and toggle, aligning with findings in (Badrinath et al. 2023). Scaling RTG values beyond attainable levels also enhances performance, consistent with observations in (Zheng, Zhang, and Grover 2022).

Ablation Results. Omitting either RTG conditioning or the probabilistic objective affects Adaptformer's adaptability to test-time cases, resulting in a $\sim$10% ↓. Qualitatively, we observed that the agent was only able to achieve a single goal in a multi-goal environment, failing to adapt to the additional goals.

Training Progression. Adaptformer, converges about 20x faster than LEAP. We hypothesize that this is due to the exclusion of image embeddings in the framework. Additionally, we utilize auxiliary mean squared losses imposed on states and reward prediction along with $L_{\text{NLL}} + L_{\text{CE}}$ to learn better state and reward representations.

Correlation between Energy and Task. Adaptformer implicitly learns the sub-goal, as shown in Fig 1. Unlike conditioning on the goals alone (without RTG), the re-scaled RTG effectively captures sub-goals tied to the task, assigning minimal energy values to crucial positions like doors and keys. The iterative planner estimates energy distributions on masked tokens (in the test case, masking all tokens within the planning horizon of $\mathbf{H}$). Action tokens are then sampled and updated based on these energy values, seeks to reduce the total energy of the planned trajectory.

Conclusion

We propose Adaptformer, which learns a novel planning heuristic and empirically show improved planning behavior, in test cases involving unseen environments, extended tasks, and auxiliary distractors.

References

Badrinath, A.; Flet-Berliac, Y.; Nie, A.; and Brunskill, E. 2023. Waypoint Transformer: Reinforcement Learning via Supervised Learning with Intermediate Targets. *ArXiv*, abs/2306.14069. 2

Chen, H.; Du, Y.; Chen, Y.; Tenenbaum, J. B.; and Vela, P. A. 2023. Planning with Sequence Models through Iterative Energy Minimization. In *International Conference on Learning Representations*. 1

Chen, L.; Lu, K.; Rajeswaran, A.; Lee, K.; Grover, A.; Laskin, M.; Abbeel, P.; Srinivas, A.; and Mordatch, I. 2021. Decision Transformer: Reinforcement Learning via Sequence Modeling. 1

Chevalier-Boisvert, M.; Bahdanau, D.; Lahlou, S.; Willems, L.; Saharia, C.; Nguyen, T. H.; and Bengio, Y. 2019. BabyAI: First Steps Towards Grounded Language Learning With a Human In the Loop. In *International Conference on Learning Representations*. 2

Emmons, S.; Eysenbach, B.; Kostrikov, I.; and Levine, S. 2022. RvS: What is Essential for Offline RL via Supervised Learning? In *International Conference on Learning Representations*. 2

Haarnoja, T.; Zhou, A.; Hartikainen, K.; Tucker, G.; Ha, S.; Tan, J.; Kumar, V.; Zhu, H.; Gupta, A.; Abbeel, P.; and Levine, S. 2018. Soft Actor-Critic Algorithms and Applications. *ArXiv*, abs/1812.05905. 2

Hung, C.-C.; Lillicrap, T.; Abramson, J.; Wu, Y.; Mirza, M.; Carnevale, F.; Ahuja, A.; and Wayne, G. 2019. Optimizing agent behavior over long time scales by transporting value. *Nature communications*, 10(1): 1–12. 1

Janner, M.; Li, Q.; and Levine, S. 2021. Offline Reinforcement Learning as One Big Sequence Modeling Problem. 1

Parisotto, E.; Song, F.; Rae, J.; Pascanu, R.; Gulcehre, C.; Jayakumar, S.; Jaderberg, M.; Kaufman, R. L.; Clark, A.; Noury, S.; et al. 2020. Stabilizing transformers for reinforcement learning. In *International conference on machine learning*, 7487–7498. PMLR. 1

Zheng, Q.; Zhang, A.; and Grover, A. 2022. Online decision transformer. In *International Conference on Machine Learning*. PMLR. 1, 2