Learning Temporally and State-Abstracted World Models for Long-Horizon Exploration

William Huang
Department of Computer Science
Yale University
New Haven, USA
william.huang@yale.edu

Benjamin Freed
Robotics Institute
Carnegie Mellon University
Pittsburgh, USA
bfreed@andrew.cmu.edu

I. Introduction

Robotic exploration often requires decision-making over long temporal horizons. In domains such as navigation and manipulation, small perturbations at the level of primitive actions rarely result in meaningful exploratory behavior. Instead, effective exploration typically emerges from temporally extended behaviors, such as traversing a doorway, reaching toward an object, or executing multi-step manipulation strategies. This observation motivates the use of temporal abstraction, where agents plan over skills or options that span multiple low-level timesteps [1]–[4].

A complementary angle focuses on state abstraction via learned world models. Rather than reasoning directly in high-dimensional observation spaces, model-based approaches learn compact latent representations that capture the essential structure for prediction and control. Dreamer-style models, for instance, learn recurrent latent dynamics that enable imagination entirely in latent space [5]. While such methods provide powerful abstractions for long-horizon reasoning, they typically rely on single-step latent transitions, limiting their ability to capture temporally extended structure.

Recent work on temporally abstract world models, such as OPOSM, decomposes offline trajectories into latent skills and learns skill-conditioned dynamics, enabling planning over sequences of skills rather than primitive actions [4]. However, these approaches operate over the original state space. Conversely, latent world models provide state abstraction but lack explicit temporal abstraction. This dichotomy raises a key question: can temporal and state abstraction be learned jointly to produce a compact, temporally abstract world model suitable for long-horizon planning?

II. Model-Based Temporal and State Abstraction

We introduce *State-Abstracted Latent skills for Temporal planning* (SALT), a framework that learns temporally extended skills and a temporally and state-abstracted world model (TSAWM) from offline trajectories. Together, these components enable planning over scored skill transitions in a learned abstract state space, reducing the effective planning horizon while preserving low-level executability.

SALT combines temporal abstraction, state abstraction, and goal-aware planning in a single offline-learned framework.

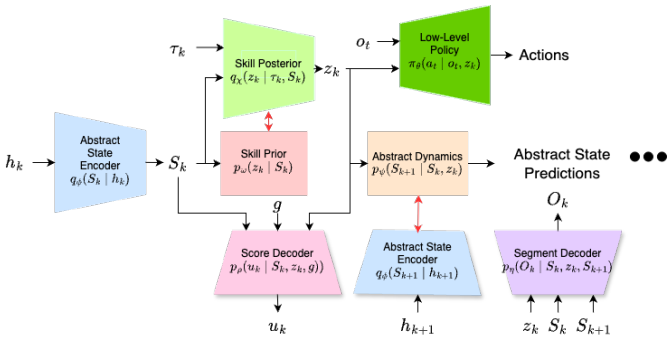

Fig. 1. **Temporally and State-Abstracted Model Architecture.** SALT infers abstract boundary states and latent skills from offline trajectory segments, predicts skill-conditioned transitions in the learned abstract state space, and scores imagined transitions for goal-conditioned planning.

Unlike OPOSM, which predicts temporally abstract transitions in the original environment state space, SALT learns a compact latent state space for planning. Unlike Dreamer-style latent world models, which predict single-step latent transitions, SALT predicts the effects of temporally extended skills over multi-step segments. Finally, SALT learns a goal-conditioned score model that assigns utility to imagined abstract transitions, enabling planning over scored latent skill sequences.

A. World Model Learning

Let an observed low-level trajectory be $o_{0:T}$, $a_{0:T-1}$, where o_t is the observation and a_t is the action. We divide the trajectory into K segments of length H, so that $T = KH$. Each segment is represented by an abstract boundary state S_k and a latent skill z_k. A skill is a temporally extended behavior primitive represented by a latent variable z_k that conditions a low-level policy $\pi_\theta(a_t \mid o_t, z_k)$ for a fixed horizon H. The abstract state S_k summarizes the trajectory prefix up to boundary k, while z_k summarizes the behavior executed over the next H low-level steps.

For segment k, define $O_k = o_{kH+1:(k+1)H}$, $A_k = a_{kH:(k+1)H-1}$, $\tau_k = (O_k, A_k)$. We propose to use a filtering posterior for abstract states and a segment-level posterior for

skills:

$$q(S_{0:K}, z_{0:K-1} \mid o_{0:T}, a_{0:T-1}) = \prod_{k=0}^{K} q_\phi(S_k \mid h_k) \prod_{k=0}^{K-1} q_\chi(z_k \mid \tau_k, S_k), \tag{1}$$

where the history available at abstract boundary k is $h_k = (o_{0:kH}, a_{0:kH-1})$.

The generative model factorizes across abstract segments:

$$\begin{aligned} p(o, a, u, S, z \mid g) &= p(S_0)p(o_0 \mid S_0) \prod_{k=0}^{K-1} \Big[p_\omega(z_k \mid S_k) p_\psi(S_{k+1} \mid S_k, z_k) \\ &\quad \cdot p_\eta(O_k \mid S_k, z_k, S_{k+1}) p_\rho(u_k \mid S_k, z_k, g) \\ &\quad \cdot \prod_{t=kH}^{(k+1)H-1} \pi_\theta(a_t \mid o_t, z_k) \Big]. \end{aligned} \tag{2}$$

Here p_ω is a state-conditioned skill prior, p_ψ is the abstract dynamics model, p_η reconstructs the observation segment, p_ρ is a goal-conditioned score decoder, and π_θ is the skill-conditioned low-level policy. The score decoder predicts a segment-level planning signal u_k, such as negative distance-to-goal in AntMaze or cumulative reward in manipulation tasks.

Maximizing the marginal likelihood yields the evidence lower bound

$$\begin{aligned} \mathcal{L} =& \mathbb{E}_q \Bigg[\log p(o_0 \mid S_0) + \sum_{k=0}^{K-1} \Big(\log p_\eta(O_k \mid S_k, z_k, S_{k+1}) \\ &\quad + \log p_\rho(u_k \mid S_k, z_k, g) \\ &\quad + \sum_{t=kH}^{(k+1)H-1} \log \pi_\theta(a_t \mid o_t, z_k) \Big) \Bigg] \\ &- D_{\mathrm{KL}} \left(q_\phi(S_0 \mid h_0) \| p(S_0) \right) \\ &- \sum_{k=1}^{K} \mathbb{E}_q \left[D_{\mathrm{KL}} \left(q_\phi(S_k \mid h_k) \| p_\psi(S_k \mid S_{k-1}, z_{k-1}) \right) \right] \\ &- \sum_{k=0}^{K-1} \mathbb{E}_q \left[D_{\mathrm{KL}} \left(q_\chi(z_k \mid \tau_k, S_k) \| p_\omega(z_k \mid S_k) \right) \right]. \end{aligned} \tag{3}$$

The reconstruction and policy terms encourage each skill to explain the low-level behavior within a segment. The abstract-state KL regularizes predicted boundary states toward the learned abstract dynamics, while the skill KL regularizes inferred skills to a compressed state-conditioned prior. The score-decoding term trains the model to assign a goal-conditioned utility to each abstract skill transition. Together, these terms produce a compact latent world model in which planning can occur over K scored abstract transitions instead of T low-level actions. The corresponding architecture diagram is shown in Fig. 1. Following OPOSM, we also use an EM-style training procedure to encourage skills to capture their causal influence on behavior. In the E-step, the skill encoder is updated toward an action-conditioned posterior that favors skills which both are likely under the prior and explain the observed actions:

$$p^\star(z_k \mid S_k, \tau_k) \propto p_\omega(z_k \mid S_k) \prod_{t=kH}^{(k+1)H-1} \pi_\theta(a_t \mid o_t, z_k). \tag{4}$$

In the M-step, the generative components are updated while holding the encoders fixed.

B. Planning with TSAWM

Once the abstract world model has been learned, the agent plans over temporally extended skills in the learned abstract state space. Given a current abstract state $\hat{S}_0$ and goal g, we use CEM to optimize a sequence of normalized skill variables $\epsilon_{0:K-1}$. Each ϵ_k is mapped to a skill through the state-conditioned prior, $z_k = \mu_\omega(\hat{S}_k) + \sigma_\omega(\hat{S}_k) \odot \epsilon_k$, $p_\omega(z_k \mid \hat{S}_k) = \mathcal{N}(\mu_\omega(\hat{S}_k), \sigma_\omega(\hat{S}_k))$. Planning in this whitened space gives a better-conditioned search problem, since likely skills correspond to values near a unit Gaussian. Candidate skill sequences are then rolled out through the abstract dynamics and scored using the goal-conditioned segment score model.

We use model predictive control where, after CEM selects a sequence, the agent executes only the first skill through the low-level policy for H steps, then re-estimates its abstract state and replans. This makes planning more robust to errors in the abstract-model.

III. Results and Conclusion

We conduct preliminary experiments on AntMaze-medium-diverse, a long-horizon offline navigation task that requires a quadruped agent to compose locomotion behaviors over many low-level timesteps. SALT is trained on fixed-length trajectory windows with segment length $H = 10$, planning horizon $K = 4$, abstract state dimension 32, and skill dimension 256. At evaluation time, the agent plans in the whitened skill space using CEM, rolls out candidate plans through the learned abstract dynamics, executes the first selected skill, and replans.

Our early SALT checkpoint yields encouraging results, achieving a success rate of approximately 71.7% for the model being trained for only 50 epochs without fine-tuning neither the world model nor the CEM planner. A representative rollout shows the agent composing temporally extended skills to move from the lower-left start region toward the goal, with each skill endpoint marking one abstract transition rather than a low-level action. These preliminary results suggest that learned abstract boundary states can preserve the benefits of temporally abstract planning while enabling search in a compact latent state space.

Future work will compare SALT against temporally abstract and low-level planning baselines on Maze2D, AntMaze-large, and Franka Kitchen. Additional qualitative and quantitative results are provided in Appendix A.

TABLE I

SUCCESS RATES FOR ANTMAZE ENVIRONMENTS. SUCCESS RATE IS MEASURED AS THE FRACTION OF EPISODES IN WHICH THE AGENT REACHES ITS GOAL. THE SALT RESULT REFLECTS AN EARLY 50-EPOCH CHECKPOINT WITHOUT MODEL OR PLANNER FINE-TUNING.

Task	SALT (Ours)	OPOSM	LLP	CQL+OPAL	BC+OPAL	BC	BEAR	EMaQ	CQL	CQL+Off-DADS	COMBO
Medium	71.7 ± 5.1	78.29 ± 4.32	31.2 ± 5.74	81.1 ± 3.1	24.0 ± 4.8	0.0	8.0	0.0	53.7 ± 6.1	59.6 ± 2.9	17.3 ± 4.3

Note: LLP denotes low-level planning [4]; OPAL denotes Offline Primitive Discovery for Accelerating Offline Reinforcement Learning [2]; BC denotes behavior cloning [6]; BEAR denotes Bootstrapping Error Accumulation Reduction [7]; EMaQ denotes Expected-Max Q-learning [8]; CQL denotes Conservative Q-Learning [9]; Off-DADS denotes offline Dynamics-Aware Discovery of Skills [10], [11]; and COMBO denotes Conservative Offline Model-Based Policy Optimization [12].

APPENDIX A
ADDITIONAL EXPERIMENTAL RESULTS

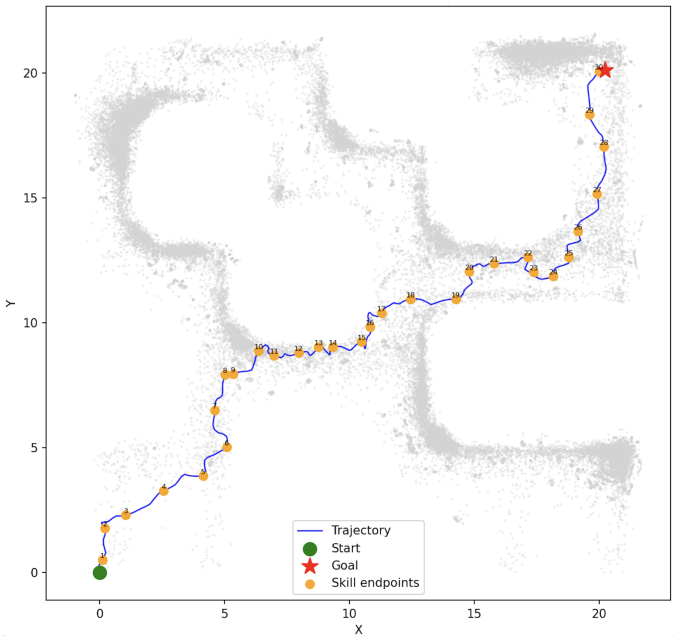

Fig. 2. Example rollout in AntMaze-medium-diverse. The blue curve shows the executed trajectory, the green circle marks the start state, the red star marks the goal, and orange points denote skill endpoints. The rollout illustrates how SALT composes temporally extended skills to navigate through the maze toward the goal.

REFERENCES

[1] R. S. Sutton, D. Precup, and S. Singh, "Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning," *Artificial Intelligence*, vol. 112, no. 1-2, pp. 181–211, 1999.

[2] A. Ajay, A. Kumar, P. Agrawal, S. Levine, and O. Nachum, "Opal: Offline primitive discovery for accelerating offline reinforcement learning," in *International Conference on Learning Representations*, 2021.

[3] K. Pertsch, Y. Lee, and J. J. Lim, "Spirl: Learning skill priors for reinforcement learning," in *International Conference on Learning Representations*, 2021.

[4] B. Freed, S. Venkatraman, G. Sartoretti, J. Schneider, and H. Choset, "Learning temporally abstract world models without online experimentation," in *International Conference on Machine Learning*, 2023.

[5] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap, "Mastering diverse domains through world models," *arXiv preprint arXiv:2301.04104*, 2023.

[6] D. A. Pomerleau, "ALVINN: An autonomous land vehicle in a neural network," in *Advances in Neural Information Processing Systems*, vol. 1, 1988.

[7] A. Kumar, J. Fu, M. Soh, G. Tucker, and S. Levine, "Stabilizing off-policy q-learning via bootstrapping error reduction," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[8] S. K. S. Ghasemipour, D. Schuurmans, and S. S. Gu, "EMaQ: Expected-max q-learning operator for simple yet effective offline and online rl," in *International Conference on Machine Learning*. PMLR, 2021, pp. 3682–3691.

[9] A. Kumar, A. Zhou, G. Tucker, and S. Levine, "Conservative q-learning for offline reinforcement learning," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 1179–1191.

[10] A. Sharma, M. Ahn, S. Levine, V. Kumar, K. Hausman, and S. Gu, "Emergent real-world robotic skills via unsupervised off-policy reinforcement learning," *arXiv preprint arXiv:2004.12974*, 2020.

[11] A. Sharma, S. Gu, S. Levine, V. Kumar, and K. Hausman, "Dynamics-aware unsupervised discovery of skills," in *International Conference on Learning Representations*, 2020.

[12] T. Yu, A. Kumar, R. Rafailov, A. Rajeswaran, S. Levine, and C. Finn, "COMBO: Conservative offline model-based policy optimization," in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 28 954–28 967.