In-Context Planning with Latent Temporal Abstractions

Anonymous authors
Paper under double-blind review

Abstract

Planning-based reinforcement learning for continuous control is bottlenecked by two practical issues: planning at primitive time scales leads to prohibitive branching and long horizons, while real environments are frequently partially observable and exhibit regime shifts that invalidate stationary, fully observed dynamics assumptions. We introduce I-TAP (In-Context Latent Temporal-Abstraction Planner), an offline RL framework that unifies in-context adaptation with online planning in a learned discrete temporal-abstraction space. From offline trajectories, I-TAP learns an observation-conditioned residual-quantization VAE that compresses each observation–macro-action segment into a coarse-to-fine stack of discrete residual tokens, and a temporal Transformer that autoregressively predicts these token stacks from a short recent history. The resulting sequence model acts simultaneously as a context-conditioned prior over abstract actions and a latent dynamics model. At test time, I-TAP performs Monte Carlo Tree Search directly in token space, using short histories for implicit adaptation without gradient updates, and decodes selected token stacks into executable actions. Across deterministic MuJoCo, stochastic MuJoCo with different latent dynamics regimes, and high-dimensional Adroit manipulation, including partially observable variants, I-TAP consistently matches or outperforms strong model-free and model-based offline baselines, demonstrating efficient and robust in-context planning under stochastic dynamics and partial observability.

1 Introduction

Planning-based reinforcement learning (RL) has delivered strong results in discrete decision-making domains (e.g., board games and video games) (Silver et al., 2017; Schrittwieser et al., 2020) and has shown increasing promise in continuous control (Hubert et al., 2021). However, planning-based offline RL for continuous control faces two practical challenges. First, many real environments are effectively partially observable due to latent parameters (e.g., unobserved disturbances or payload changes), which breaks the stationary, full observability assumptions commonly adopted by learned dynamics models used for planning. When these latent factors are not properly handled during planning, they manifest as apparent stochasticity from the planner's perspective (Antonoglou et al., 2022), exacerbating planning complexity. Second, planning over primitive continuous actions induces large branching factors and long effective horizons, making search expensive especially under uncertainty.

To address adaptation under partial observability, recent work has reframed RL as sequence modeling, leveraging Transformers (Vaswani et al., 2017) trained on trajectories to produce policies conditioned on a finite context window (Chen et al., 2021; Brown et al., 2020; Furuta et al., 2022; Liu & Abbeel, 2023; Laskin et al., 2023; Huang et al., 2024). Conditioning on recent interaction history enables in-context adaptation without test-time gradient updates, and can implicitly capture latent task variables when they are identifiable from history. Yet these sequence-model policies are typically deployed as direct action predictors, which introduces two limitations: (i) Without an explicit decision-time optimizer, they often inherit the constraints and suboptimalities of the offline dataset (Son et al., 2025). (ii) In stochastic environments, converting a predictive model into an optimized decision rule is nontrivial when the model is used only as a conditional policy (Paster et al., 2022). These issues motivate combining in-context models with explicit planning.

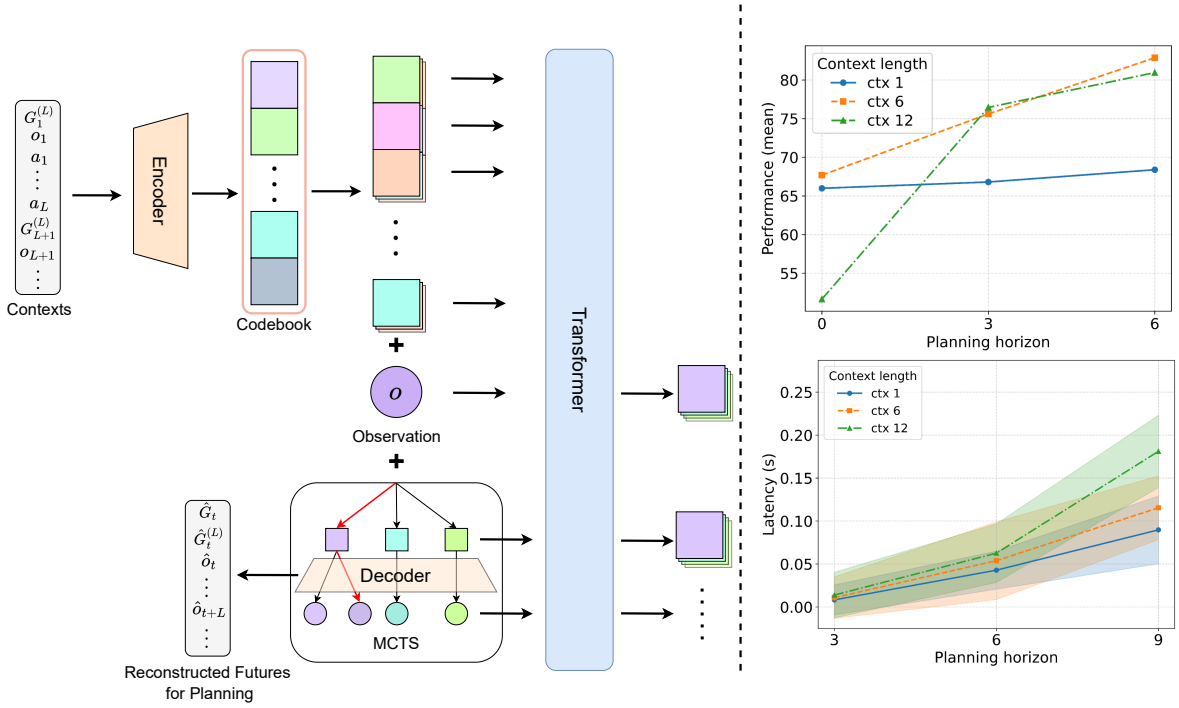


Figure 1: Overview of I-TAP. **Left:** A residual-quantized VAE (RQ-VAE) discretizes continuous observation–action trajectories into a coarse-to-fine token stack. **Right:** Normalized return and per-decision latency as functions of planning horizon and context size on Stochastic MuJoCo, highlighting the importance of a properly sized context window for effective in-context planning under environmental stochasticity and partial observability.

Nevertheless, planning directly in raw continuous action space can be inefficient and inflexible (Jiang et al., 2023). Recent planning-based methods reduce complexity by learning temporal abstractions (e.g., options (Sutton et al., 1999) or macro-actions (Mcgovern & Sutton, 1998)) and planning over high-level decisions (Jiang et al., 2023; Luo et al., 2025), which shortens effective horizons and reduces branching. However, these planners are developed under the assumption of a stationary, fully observed Markov decision process (MDP) and learn dynamics models that condition only on the current state. As a result, state aliasing would create apparent stochasticity and increase the search burden.

Motivated by these challenges, we introduce an in-context planning framework that adapts to different latent parameters in a learned discrete latent temporal abstraction space for continuous control under stochastic dynamics. Our premise is that integrating temporal abstraction with in-context adaptation for planning-based RL addresses these issues jointly. Adapting from recent history and planning conditioned on it in a latent temporal abstraction space allows an agent to: (i) decouple adaptation and planning from the native temporal granularity of MDP, thereby shortening the required context, easing the learning of a reliable sequence model prior by modeling a simpler distribution over discrete latent temporal abstractions instead of high-dimensional continuous actions, and reducing the branching factor during planning; (ii) use context to infer the environment's dynamics (e.g., unobserved perturbation forces), enabling effective adaptation and forecasting across scenario shifts; and (iii) employ an online planner such as Monte Carlo Tree Search (MCTS) for optimization, providing a mechanism to deviate from suboptimal behavior policies in the offline data and handle uncertainty.

To this end, we propose *In-Context Latent Temporal Abstraction Planner* (I-TAP), which learns a discrete latent temporal-abstraction space and a Transformer-based in-context sequence prior over discrete latent codes from offline data, and performs decision-time planning with these models to enable adaptive decision-

making. To improve flexibility and scalability in handling high-dimensional continuous observation-action spaces, we adopt an observation-conditioned residual-quantized VAE (RQ-VAE) to learn this discrete latent temporal-abstraction space, as illustrated in Fig. 1. RQ-VAE (Lee et al., 2022) encodes each macro-step into a depth-D coarse-to-fine stack of code indices drawn from a codebook, providing a compositional discretization with substantially higher effective capacity than a single code. This retains high-fidelity decoding while enabling compact discrete representations, yielding a scalable interface for in-context sequence modeling over abstractions and planning in latent space. MCTS then operates over these latent tokens using context-guided priors to balance exploration and exploitation under uncertainty; finally, we decode the selected latent stack to a primitive action sequence and execute the first action.

Our experiments demonstrate that I-TAP can be trained as a single offline model across behavior policies of varying quality and multiple latent parameters, and evaluated from deterministic to highly stochastic environments. Across these settings, I-TAP matches or outperforms strong offline RL and planning-based baselines, while exhibiting in-context adaptation under partial observability and scaling to high-dimensional continuous control through latent temporal abstractions. In summary, we make the following contributions: (i) We propose I-TAP, an offline RL method that unifies in-context adaptation and online planning for stochastic, partially observable continuous control by planning in a learned latent temporal-abstraction space; (ii) To improve scalability and flexibility, we learn a residual-quantized temporal abstraction and a context-conditioned latent dynamics model from offline trajectories, via an observation-conditioned RQ-VAE and an autoregressive temporal Transformer; (iii) We instantiate an in-context planner that performs Monte Carlo Tree Search directly over latent tokens, reducing the effective branching factor and decision horizon, and enabling improvements beyond suboptimal behavior policies.

2 Problem Formulation

POMDP. We consider a partially observable Markov decision process (POMDP) $M = (\mathcal{S}, \mathcal{A}, \mathcal{O}, P, R, \gamma)$, where $P(s_{t+1} \mid s_t, a_t)$ is the transition kernel, $R(r_t \mid s_t, a_t)$ is the reward distribution, and $\gamma \in [0, 1)$ is the discount factor. At each time step t, the agent receives an observation $o_t \in \mathcal{O}$, selects an action $a_t \in \mathcal{A}$, the environment transitions according to $P(s_{t+1} \mid s_t, a_t)$, and the agent receives a reward $r_t \sim R(\cdot \mid s_t, a_t)$. For a fixed context length c, we define the agent's context prior to observing o_t as $c_t = (o_{t-c}, a_{t-c}, r_{t-c}, \ldots, o_{t-1}, a_{t-1}, r_{t-1}) \in \mathcal{C}$. The goal is to maximize the expected discounted return $J = \mathbb{E}_{\pi,M}\left[\sum_{t=0}^{T-1} \gamma^t r_t\right]$ over horizon T.

Meta-RL. We define a context-conditioned policy $\pi : \mathcal{C} \times \mathcal{O} \to \Delta(\mathcal{A})$, where $\mathcal{C}$ is a fixed-length context space and $\Delta(\mathcal{A})$ denotes distributions over actions. At time t, the algorithm forms a bounded context $c_t \in \mathcal{C}$ from the interaction history h_t (e.g., a sliding window of recent transitions) and samples actions as $a_t \sim f(c_t, o_t)$. Meta-RL aims to learn an algorithm π that maximizes J in expectation over a distribution of tasks (POMDPs) $p(M)$: $\max_\pi \ \mathbb{E}_{M\sim p(M)}\left[\mathbb{E}_{\pi,M}\left[\sum_{t=0}^{H-1} \gamma^t r_t\right]\right]$. In our setting, each M is induced by an unobserved latent parameter. In offline meta-RL, we assume access to a dataset of trajectories collected by some behavior algorithms on meta-training tasks, from which the context c_t is constructed.

3 Background

Residual-Quantized Variational Autoencoder (RQ-VAE). Residual-Quantized VAE (RQ-VAE) (Lee et al., 2022) is a discrete autoencoding model that replaces the single-step vector quantization in VQ-based models with residual quantization, representing each latent as a stack of D discrete codes. Let $z_t \in \mathbb{R}^d$ denote a continuous feature vector produced by an encoder, and let the codebook be $E = \{e_1, \ldots, e_K\} \subset \mathbb{R}^d$. RQ-VAE iteratively quantizes residuals in a coarse-to-fine manner by initializing $r_t^{(0)} := z_t$ and, for depths $\ell = 1, \ldots, D$, computing

$$k_{t,\ell} = \arg\min_{k\in[K]} \|r_t^{(\ell-1)} - e_k\|_2^2, \qquad r_t^{(\ell)} = r_t^{(\ell-1)} - e_{k_{t,\ell}},$$

which yields the depth-ℓ partial sum $\hat{z}_t^{(\ell)} := \sum_{j=1}^{\ell} e_{k_{t,j}}$ and the final quantized vector $\hat{z}_t := \hat{z}_t^{(D)}$. Using a shared codebook across depths yields an effective representational capacity that grows as K^D without enlarging K. The decoder reconstructs the input from $\hat{z}_t$, and the model is trained jointly with minimizing a reconstruction loss plus a depth-wise commitment regularizer, using the straight-through estimator for backpropagation through discrete assignments and updating the codebook via exponential moving average (EMA) (van den Oord et al., 2017). An RQ-Transformer (Lee et al., 2022) combines a spatial Transformer that summarizes codes from previous positions into a context vector with a depth Transformer that, conditioned on this context, predicts the D codes within the current stack in a coarse-to-fine manner, thereby modeling an autoregressive prior over code stacks.

Temporal Abstraction via Macro-Actions. We introduce temporal abstractions via a fixed macro length L and a macro-action space $\mathcal{A}^{(L)}$, where each macro-action $m_b \in \mathcal{A}^{(L)}$ consists of L primitive actions. Decisions are made at macro-steps $b = 0, 1, \ldots, N-1$, corresponding to primitive times $t = bL$, so the episode horizon satisfies $T = NL$. At macro-step b, the agent selects m_b and executes it for L primitive steps, generating rewards $\{r_{bL}, \ldots, r_{bL+L-1}\}$.

Trajectory Representation. Consider an episode with $T = NL$ primitive steps and macro boundaries at $t = bL$. The return-to-go from primitive time t is $G_t = \sum_{i=t}^{T-1} \gamma^{i-t} r_i$, and the L-step discounted return from a macro boundary is $G_{bL}^{(L)} = \sum_{i=0}^{L-1} \gamma^i r_{bL+i}$. We write the macro-level trajectory as $\tau = \big((G_{bL}, G_{bL}^{(L)}, o_{bL}, m_b)\big)_{b=0}^{N-1}$, where o_{bL} is the observation at the macro boundary and m_b is the macro-action executed for the subsequent L primitive steps. This representation preserves the underlying primitive dynamics while exposing temporally extended actions.

4 Method

In this section, we provide an in-depth explanation of each component of I-TAP: the discretization of observation-macro-actions sequences, modeling the in-context prior and transition distribution over latent codes, and the planning process with MCTS. In general, I-TAP is a generative model for trajectories conditioned on both current observation and recent historical context, allowing for efficient in-context planning and decision-making.

4.1 In-Context Residual Discretization of Observation Macro Action Sequences

In-context RL approaches tend to replicate the suboptimal behaviors of the source algorithm (Son et al., 2025), necessitating the integration of a planning mechanism to deviate from suboptimal decisions. Meanwhile, a discrete action space simplifies the representation of action distributions and facilitates the use of advanced planning algorithms (Silver et al., 2018).

To leverage these advantages, prior work has proposed state-conditioned Vector Quantized VAEs (VQ-VAEs) (Jiang et al., 2023) to discretize continuous control into a compact latent action space. However, single-code vector quantization can become a bottleneck: to maintain reconstruction fidelity for high-dimensional features, one often needs either a very large codebook (inflating the discrete vocabulary) or additional compression, both of which can degrade reconstruction quality when features are high-dimensional (Lee et al., 2022; Jiang et al., 2024). In our setting, each macro-step token must summarize an observation together with a temporally extended action, and the encoder processes these as context-dependent sequences, making the embeddings more context-dependent and diverse, which further increases the required quantization capacity. We therefore adopt an observation-conditioned RQ-VAE, which represents each macro-step with a depth-D coarse-to-fine stack of codes, providing substantially higher effective capacity without requiring an excessively large single-step codebook. This yields compact discrete representations while preserving decoding fidelity, which is crucial when downstream decision making depends on model predictions over these abstractions.

Tokens and masking. At macro index b, a token is $x_{t_b} = \big(G_{t_b}, G_{t_b}^{(L)}, o_{t_b}, m_{t_b}\big)$. During training, we sample contiguous chunks $(x_{t_{b-c}}, x_{t_{b-c+1}}, \ldots, x_{t_b}, x_{t_{b+1}})$, where the first c tokens provide context and the last two tokens correspond to the current and next macro-step.

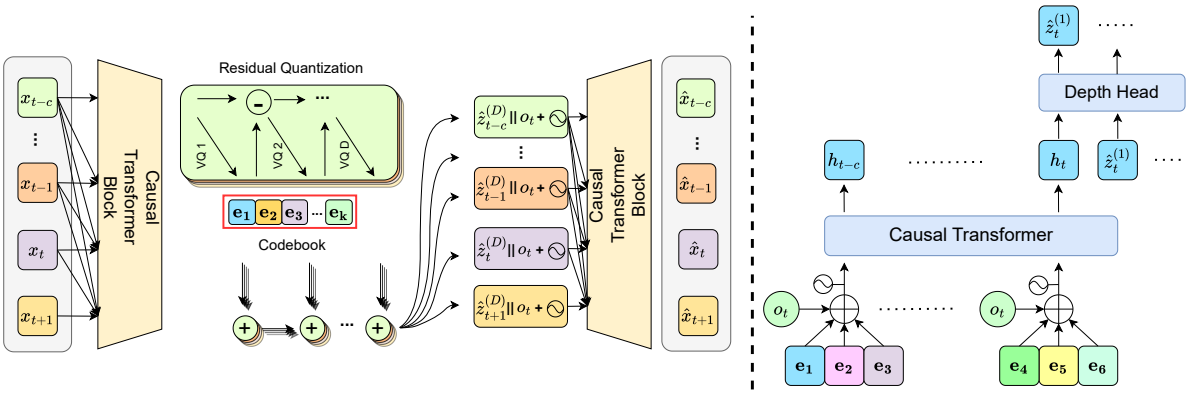


Figure 2: An overview of our RQ-VAE model that discretizes state-macro action sequences and temporal prior for I-TAP

Inspired by Luo et al. (2025), we mask return components to reduce variance induced by behavior-policy quality and environmental stochasticity. Specifically, we mask the return-to-go G_t at all positions, since it is a high-variance long-horizon signal and requires choosing a test-time target value. We retain the short-horizon return $G_t^{(L)}$ only in the context tokens and mask it for the current and next tokens $(x_{t_b}, x_{t_{b+1}})$. Consequently, the code assigned to the current macro decision is determined by the observation–macro-action pair and the preceding context, rather than by the realized return, which reduces sensitivity to return variability in the offline data.

This design avoids manually choosing a target G_{t_b}, which is susceptible to luck-induced variability in stochastic environments (Paster et al., 2022), while still using $G^{(L)}$ in the context as a stable short-horizon signal for inferring latent regimes.

Discretization with RQ-VAE A causal Transformer maps the chunk to per-token features $Z = (z_{t_{b-c}}, \ldots, z_{t_b}, z_{t_{b+1}})$. For each token index $t \in \{t_b - c, \ldots, t_b + 1\}$, we apply residual quantization (section 3) with a shared codebook $E = \{e_1, \ldots, e_K\} \subset \mathbb{R}^d$ to obtain a depth-D code stack $k_{t,1:D}$ and partial sums $\hat{z}_t^{(\ell)} = \sum_{j=1}^{\ell} e_{k_{t,j}}$; we use the final quantized embedding $\hat{z}_t^{(D)}$ as the discrete representation of x_t.

Motivated by evidence from prior work (Jiang et al., 2023; Luo et al., 2023) that state-conditioned decoding allows more compact codebooks while preserving reconstruction fidelity. For each token, our decoder conditions on o_{t_b} and the quantized latent $\hat{z}_t^{(D)}$ via a linear adapter and causal attention, reconstructing all features for every token in the chunk by leveraging its previous context: $(\hat{G}_{t_{b-c}}, \hat{G}^{(L)}_{t_{b-c}}, \hat{o}_{t_{b-c}}, \hat{m}_{t_{b-c}}, \ldots, \hat{G}_{t_{b+1}}, \hat{G}^{(L)}_{t_{b+1}}, \hat{o}_{t_{b+1}}, \hat{m}_{t_{b+1}})$. We optimize a reconstruction loss plus a residual partial-sum commitment:

$$\mathcal{L} = \sum_{\tau} \alpha_\tau \big\|(\hat{x}_\tau - x_\tau)\big\|_2^2 + \frac{\beta_{\text{ps}}}{D} \sum_{\ell=1}^{D} \big\|Z - \text{sg}\Big[\hat{Z}^{(\ell)}\Big]\big\|_2^2.$$

Here $\alpha_\tau = \alpha_{\text{tail}}$ for the last two tokens $(x_{t_b}, x_{t_{b+1}})$ and α_{ctx} otherwise; $\hat{Z}^{(\ell)}$ is the depth-wise ℓ partial sum of residual code embeddings; and $\text{sg}[\cdot]$ denotes stop-gradient. The depth-wise partial-sum term stabilizes residual quantization and prevents code hopping across depths, consistent with RQ-VAE (Lee et al., 2022).

4.2 Temporal Prior over Residual Code Stacks

Let $k_{t,1:D}$ denote the depth-1:D codes at macro time t. We learn an observation conditioned depth-aware autoregressive prior that factorizes across time and within-time depth:

$$p_\phi\big(k_{t,1:D} \,\big|\, k_{<t,1:D},\, o_t\big) \;=\; \prod_{\ell=1}^{D} p_\phi\big(k_{t,\ell} \,\big|\, k_{<t,1:D},\, k_{t,<\ell},\, o_t\big).$$

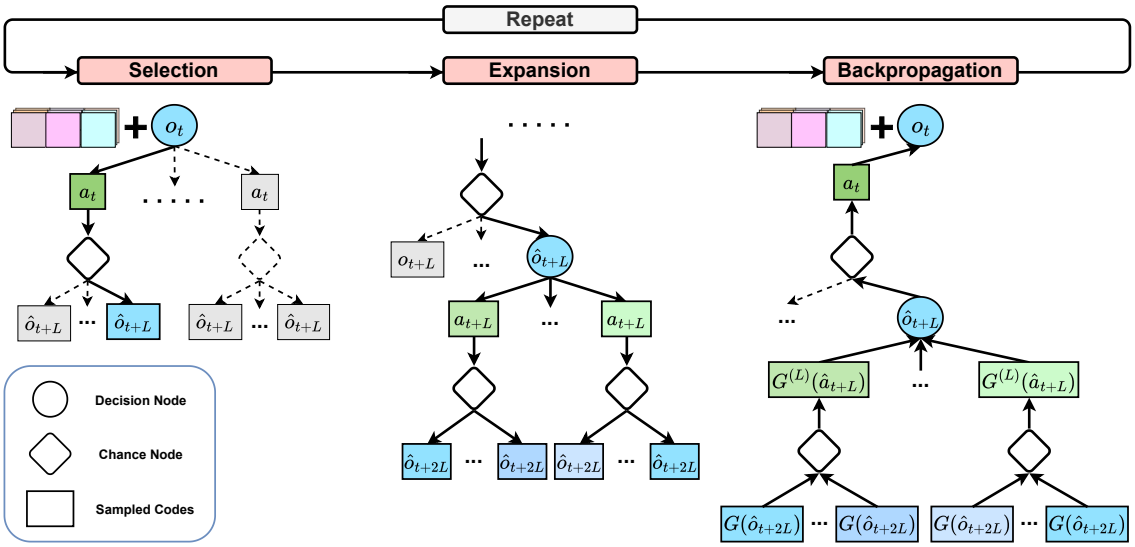


Figure 3: Macro-level MCTS overview. Each iteration uses P-UCT to select a macro-action, expands several candidates and their predicted outcomes in parallel, then backs up the resulting Q-estimates through the search tree to steer subsequent exploration.

Inspired by the factorized residual-code prior in RQ-Transformer (Lee et al., 2022), we design an efficient prior for planning that avoids expanding a depth-D code stack into D separate tokens per time step (which would increase context length from C to CD). Specifically, a temporal trunk (causal over t) embeds each past position by the sum of its depth embeddings $\sum_{j=1}^{D} e_{k_{u,j}}$, adds positional and observation embeddings, and produces a context h_t. We use a lightweight depth head (a shared Multi-Layer Perceptron) to predict $k_{t,1}, k_{t,2}, \ldots, k_{t,D}$ by conditioning on h_t, a depth embedding, and the partial sum of shallower depths at time t. We then minimize the negative log-likelihood loss:

$$\mathcal{L}_{\text{prior}} \;=\; \mathbb{E}\Big[-\sum_{t}\sum_{\ell=1}^{D} \log p_{\phi}\big(k_{t,\ell} \,\big|\, k_{<t,1:D},\, k_{t,<\ell},\, o_t\big)\Big].$$

The time and depth factorization retains long temporal context while modeling within-position coarse to fine refinement efficiently.

4.3 In-Context Planning With Monte Carlo Tree Search

Prior work has leveraged MCTS to mitigate stochasticity arising from the environment in both online (Antonoglou et al., 2022) and offline RL (Luo et al., 2025). Planning in the real world, however, poses two additional challenges. First, partial observability induces apparent stochasticity when the context cannot reliably disambiguate latent states. Second, one needs a mechanism to balance the inherited bias and exploration at decision time when policies are learned from suboptimal behavior.

We therefore adopt MCTS as the online planner in our latent temporal-abstraction space (Fig. 4.3). Using the learned latent dynamics to sample multiple future continuations and backing up their returns, MCTS estimates the expected value of each candidate latent action, making action selection less sensitive to noisy single-rollout return estimates. Moreover, by planning over a context-conditioned latent state, the search mitigates apparent stochasticity from partial observability (to the extent captured by the context). Finally, coupling P-UCT (Silver et al., 2017) with our context-conditioned prior over latent tokens enables targeted exploration and provides a principled way to override suboptimal priors when predicted returns justify it, while keeping the search within the distribution supported by the learned model.

Latent Search Tree. At time t, the agent observes o_t and an interaction history in a sliding window of length $L \times c$. We encode this history into residual-quantized codes to obtain a context window $k_{t-1:t-c,\,1:D}$. A *decision node* is $s = (o_t, k_{t-1:t-c,1:D})$, and an *action edge* a out of s is a depth D code stack $k_{t,1:D}$. Executing a produces a distribution over *outcome codes* $k_{t+1,1:D}$ via our temporal prior $p_\phi(k_{t+1,1:D} \mid k_{t,1:D}, o_t, k_{t-1:t-c,1:D})$, and each outcome is decoded to a tail $(\hat{G}_{t+L}, \hat{G}^{(L)}_{t+L}, \hat{o}_{t+L}, \hat{m}_{t+L})$, yielding the successor decision node s_{t+L}. To mitigate the cost of iterative model calls with context (a bottleneck when parallelism is underused), following Luo et al. (2025) we pre-construct a context-conditioned latent search tree by parallel sampling and caching a finite set of high-probability action stacks and their predicted outcomes from p_ϕ, and run MCTS over this cached search space (section A.1).

Policy-guided selection From node s, we restrict top-K action candidates and use a behavior-like prior to prioritize searching in-distribution actions without sacrificing exploration. Let the policy head produce probability logits l_a, we select a by the AlphaZero-style P-UCT score:

$$a = \arg\max_a \Big[Q(s,a) \; + \; \underbrace{\Big(c_1 + \log \frac{N(s)+c_2+1}{c_2}\Big) \frac{\sqrt{N(s)}}{1+N(s,a)}}_{\text{exploration term}} \; \underbrace{\Big(\pi_T(a \mid s) = \frac{e^{l_a/T}}{\sum_{b \in \mathcal{A}(s)} e^{l_b/T}}\Big)}_{\text{temperature scaled prior}} \Big],$$

where $N(s)$ and $N(s,a)$ are visit counts and $Q(s,a)$ is the action value; $c_1, c_2 > 0$ are P-UCT exploration constants. This coupling of a learned prior with P-UCT emphasizes in-distribution actions (large π_T) without sacrificing exploration via the exploration term, allowing the search to deviate from the source policy when returns warrant it.

Parallel expansion and backpropagation. At each decision node s_t we expand the top-K candidates in parallel, sample D outcome codes at the chance node, and decode them to obtain a successor $s_{t+L} = (\hat{o}_{t+L},\, k'_{t:t-c+1,\,1:D})$ and its leaf value. We then back up $Q(s,a)$ along the visited path using incremental averages and update visit counts.

5 Experiments

We evaluate I-TAP through comprehensive empirical studies using tasks from the D4RL benchmark (Fu et al., 2020), focusing on standard Gym locomotion tasks and complex high-dimensional Adroit robotic manipulation. Our experiments assess both performance and adaptability of I-TAP across varying degrees of environmental stochasticity. We additionally conduct ablation studies to examine how macro-action length, context length, planning horizon, and residual depth affect performance. We further analyze the relationship between decision latency and context length in Section 5.3.

Baselines. We compare I-TAP to strong offline RL baselines: model-free actor–critic methods Conservative Q-Learning (CQL; Kumar et al. (2020)) and Implicit Q-Learning (IQL; Kostrikov et al. (2022)); context-conditioned policy methods such as Decision Transformer (DT; Chen et al. (2021)); and model-based planners that operate over learned temporal abstractions, including Trajectory Autoencoding Planner (TAP; Jiang et al. (2023)), Latent Macro-Action Planner (L-MAP; Luo et al. (2025)), where TAP is largely insensitive to raw action dimensionality and shows strong performance on high-dimensional Adroit manipulation, and L-MAP likewise scales well to high-dimensional control while remaining robust under stochastic dynamics. Finally, a risk-sensitive, model-based specialist for stochastic domains (1R2R; Rigter et al. (2023)).

Experimental Setup. To assess I-TAP's adaptation capabilities across varying latent task parameters and associated dynamics, we conduct comprehensive experiments using the Stochastic MuJoCo tasks introduced by Rigter et al. (2023). Each environment defines an unobserved global latent task parameter controlling perturbation levels (0, 2.5, 5), which in turn specify the distribution of instantaneous hidden forces at each step. We train a single model per method (I-TAP, DT, L-MAP) with three random seeds on the pooled data across perturbation regimes for each behavior-policy dataset type. At test time, I-TAP and DT perform no gradient updates; adaptation arises only through conditioning on the context window. We also train L-MAP on the same pooled data as a planning baseline that is robust to regime variation but does not rely on in-context adaptation. For standard offline RL baselines (e.g., CQL/IQL), we compare the methods with

dataset-specific models. To test scalability to high-dimensional control and partial observability, we evaluate on Adroit (Rajeswaran et al., 2018) in (i) the original fully observable setting and (ii) a partially observable variant where we mask a subset of target-position coordinates (see Appendix A.2 for the exact dimensions and masking procedure). Unless otherwise noted, we set the latent context size to $C = 6$ tokens; with a macro length of $L = 3$, this window summarizes 18 past primitive transitions. Further domain and hyperparameter details appear in Appendix A.2.

5.1 Main results

Table 1: Normalised results for high-noise (5), moderate-noise (2.5), deterministic (0) environments. Bold numbers indicate the best scores in each row.

		Model-Based				Model-Free		
Dataset Type	**Env (Noise)**	**I-TAP**	**L-MAP**	**TAP**	**1R2R**	**DT**	**CQL**	**IQL**
Medium-Expert	Hopper (5)	**82.87 ± 3.55**	71.49 ± 3.46	37.31 ± 3.66	37.99 ± 2.71	61.87 ± 2.56	68.03 ± 3.94	44.83 ± 2.58
	Hopper (2.5)	104.45 ± 2.66	93.40 ± 3.65	40.86 ± 5.42	52.19 ± 8.37	73.86 ± 2.68	**106.17 ± 2.16**	60.61 ± 3.46
	Hopper (0)	**111.71 ± 0.08**	106.74 ± 2.24	85.55 ± 3.83	57.40 ± 6.06	101.6 ± 1.85	105.4	91.5
	Walker2D (12)	**93.50 ± 3.15**	92.75 ± 1.34	91.09 ± 2.78	32.38 ± 4.55	52.42 ± 1.27	83.18 ± 3.70	68.61 ± 3.33
	Walker2D (7)	**97.29 ± 1.68**	93.48 ± 1.20	91.40 ± 1.42	56.48 ± 7.51	64.67 ± 1.00	91.44 ± 1.44	86.66 ± 1.84
	Walker2D (0)	101.20 ± 1.91	100.38 ± 0.72	105.32 ± 2.03	73.18 ± 6.29	64.65 ± 0.79	108.8	**109.6**
Mean (Medium-Expert)		**98.50**	93.04	75.26	51.60	69.85	93.84	76.97
Medium	Hopper (5)	**67.80 ± 2.60**	59.05 ± 2.93	43.93 ± 2.66	33.99 ± 0.92	55.91 ± 2.02	45.21 ± 2.97	49.69 ± 2.47
	Hopper (2.5)	**72.47 ± 2.71**	63.21 ± 3.10	43.64 ± 2.25	65.24 ± 3.31	60.97 ± 0.82	49.92 ± 3.00	56.00 ± 3.60
	Hopper (0)	**81.94 ± 2.14**	61.65 ± 2.81	69.14 ± 2.33	55.49 ± 3.99	58.14 ± 0.24	58.0	66.3
	Walker2D (12)	60.35 ± 2.73	59.05 ± 2.30	52.20 ± 2.76	32.13 ± 4.51	32.20 ± 0.83	**61.49 ± 3.24**	47.53 ± 3.05
	Walker2D (7)	**65.85 ± 2.44**	62.23 ± 1.84	44.46 ± 1.82	65.16 ± 2.84	43.77 ± 0.95	49.38 ± 2.02	48.82 ± 2.31
	Walker2D (0)	**79.57 ± 1.22**	75.54 ± 1.59	51.75 ± 3.30	55.69 ± 4.97	55.36 ± 0.61	72.5	78.3
Mean (Medium)		**71.33**	63.46	50.85	51.28	51.06	56.08	57.77
Medium-Replay	Hopper (5)	**70.67 ± 2.59**	60.76 ± 2.79	48.69 ± 2.97	68.25 ± 3.78	35.17 ± 0.96	51.70 ± 3.09	43.27 ± 2.78
	Hopper (2.5)	**81.33 ± 2.19**	73.81 ± 2.67	38.10 ± 3.22	22.82 ± 2.08	35.76 ± 1.01	40.53 ± 1.52	49.12 ± 3.38
	Hopper (0)	86.57 ± 2.03	90.8 ± 0.63	80.92 ± 3.79	89.67 ± 1.92	43.01 ± 1.36	**95.0**	94.7
	Walker2D (12)	**70.60 ± 3.07**	59.16 ± 2.92	55.15 ± 3.29	65.63 ± 3.41	37.22 ± 0.78	50.33 ± 3.88	45.13 ± 2.38
	Walker2D (7)	**74.04 ± 1.94**	69.20 ± 2.55	43.49 ± 2.27	52.23 ± 2.22	49.51 ± 0.81	40.24 ± 1.67	40.77 ± 2.72
	Walker2D (0)	75.49 ± 2.52	70.66 ± 1.78	72.32 ± 3.26	**90.67 ± 1.98**	48.44 ± 0.76	77.2	77.2
Mean (Medium-Replay)		**76.45**	70.73	56.45	64.88	41.52	59.17	58.37

Mujoco We use the Stochastic MuJoCo suite to evaluate I-TAP's in-context adaptation and robustness to uncertainty. Each episode is governed by a latent task parameter that selects the perturbation regime (e.g., 0, 2.5, 5) and thereby determines the distribution of hidden forces. We therefore report results in two complementary settings. First, Table 1 evaluates *in-distribution* regimes where latent-parameter values that are present in the meta-training datasets so that performance primarily reflects a method's ability to infer the active regime from history, adapt its decisions online, and handle uncertainty. Second, Table 2 evaluates held-out regimes not included in training to evaluate interpolation and extrapolation across latent task parameters.

Across all dataset types, I-TAP achieves the highest mean score and remains strong across changing dynamics without any gradient updates at test time. These gains are consistent with I-TAP's ability to (i) leverage trajectory history for in-context identification of the latent regime and (ii) plan online to optimize actions instead of merely imitating the behavior policy in the dataset. Relative to L-MAP, I-TAP's improvements highlight the benefit of *context-guided* action selection within search: the sequence-model prior steers exploration while the planner evaluates alternatives, rather than relying on MCTS alone to handle uncertainty. In contrast, DT degrades more sharply under stochastic dynamics and lower-quality behavior data, consistent with its lack of an integrated planner and its reliance on return conditioning, which can be sensitive to luck-induced variability (Paster et al., 2022). By combining context-based adaptation with downstream planning, I-TAP is less likely to inherit suboptimal dataset behavior and can deviate whenever the planner identifies more promising actions.

Adroit Control Adroit poses high-dimensional state–action spaces and fine-grained control demands. Table 3 shows that I-TAP achieves the best performance in both fully observable and partially observable setting. On cloned datasets, I-TAP outperforms both L-MAP and TAP in three of four tasks, indicating that planning with

Table 2: Normalised results (mean ± std). Noise parameter values are shown in parentheses. Bold indicates the best score in each row.

Dataset Type	Env (Noise)	I-TAP	L-MAP	DT
Medium-Expert	Hopper (7.5)	**70.43 ± 3.17**	62.90 ± 3.50	58.96 ± 0.21
	Hopper (3.75)	**85.07 ± 3.57**	80.51 ± 3.86	64.97 ± 0.25
	Hopper (1.25)	**106.91 ± 2.17**	94.65 ± 3.03	87.41 ± 0.24
	Walker2D (15)	**88.26 ± 3.67**	87.21 ± 3.42	44.07 ± 0.26
	Walker2D (9.5)	**94.29 ± 1.97**	92.31 ± 1.81	56.23 ± 0.22
	Walker2D (3.5)	**98.80 ± 1.00**	96.53 ± 0.62	65.07 ± 0.18
Mean (Medium-Expert)		**90.63**	85.69	62.79
Medium	Hopper (7.5)	**60.88 ± 2.49**	55.26 ± 2.53	51.62 ± 0.16
	Hopper (3.75)	**71.43 ± 2.75**	66.68 ± 2.50	56.09 ± 0.17
	Hopper (1.25)	**74.60 ± 2.46**	71.08 ± 2.56	59.39 ± 0.12
	Walker2D (15)	**54.60 ± 2.52**	50.10 ± 2.82	30.69 ± 0.18
	Walker2D (9.5)	**60.22 ± 2.49**	55.96 ± 2.53	39.16 ± 0.19
	Walker2D (3.5)	**76.56 ± 1.58**	75.77 ± 1.59	51.89 ± 0.17
Mean (Medium)		**66.38**	62.47	48.14
Medium-Replay	Hopper (7.5)	**56.96 ± 3.07**	50.79 ± 2.56	30.17 ± 0.14
	Hopper (3.75)	**75.22 ± 2.72**	70.06 ± 2.87	33.03 ± 0.18
	Hopper (1.25)	**84.43 ± 2.09**	83.99 ± 2.21	39.22 ± 0.24
	Walker2D (15)	**60.70 ± 3.11**	50.64 ± 2.87	29.91 ± 0.15
	Walker2D (9.5)	**71.90 ± 2.22**	64.68 ± 2.71	42.78 ± 0.16
	Walker2D (3.5)	**75.11 ± 1.86**	71.98 ± 2.12	58.49 ± 0.15
Mean (Medium-Replay)		**70.72**	65.36	38.93

Table 3: Adroit robotic hand control results.

Dataset Type	**Env**	**I-TAP**	**L-MAP**	**TAP**
Cloned	Pen	**85.44 ± 8.19**	60.68 ± 7.88	46.44 ± 7.54
Cloned	Hammer	**4.38 ± 1.28**	2.43 ± 0.29	1.32 ± 0.12
Cloned	Door	**14.17 ± 1.34**	13.22 ± 1.34	13.45 ± 1.43
Cloned	Relocate	0.08 ± 0.02	**0.15 ± 0.13**	-0.23 ± 0.01
Expert	Pen	**133.81 ± 5.23**	126.60 ± 5.60	127.40 ± 7.70
Expert	Hammer	**128.37 ± 0.21**	127.16 ± 0.29	127.60 ± 1.70
Expert	Door	**105.98 ± 0.08**	105.24 ± 0.10	104.80 ± 0.80
Expert	Relocate	**109.85 ± 0.88**	107.57 ± 0.76	106.21 ± 1.61
Expert (POMDP)	Pen	**81.68 ± 9.83**	69.84 ± 9.81	60.87 ± 9.55
Expert (POMDP)	Hammer	**72.36 ± 8.48**	59.21 ± 6.52	42.22 ± 12.92
Expert (POMDP)	Door	**95.05 ± 3.12**	89.35 ± 3.41	83.71 ± 4.22
Expert (POMDP)	Relocate	**52.16 ± 4.46**	37.36 ± 3.84	33.94 ± 3.50
Mean (Expert)		**119.50**	116.64	116.50
Mean (Expert POMDP)		**75.31**	63.94	55.19

context-guided prior in the latent abstraction space enables the agent to do more targeted exploration with promising returns while remaining in distribution. On expert datasets, I-TAP matches or exceeds baselines, and under partial observability (POMDP) it maintains the top scores across all four tasks, highlighting the benefit of context-conditioned planning when state aliasing induces apparent stochasticity and its ability to

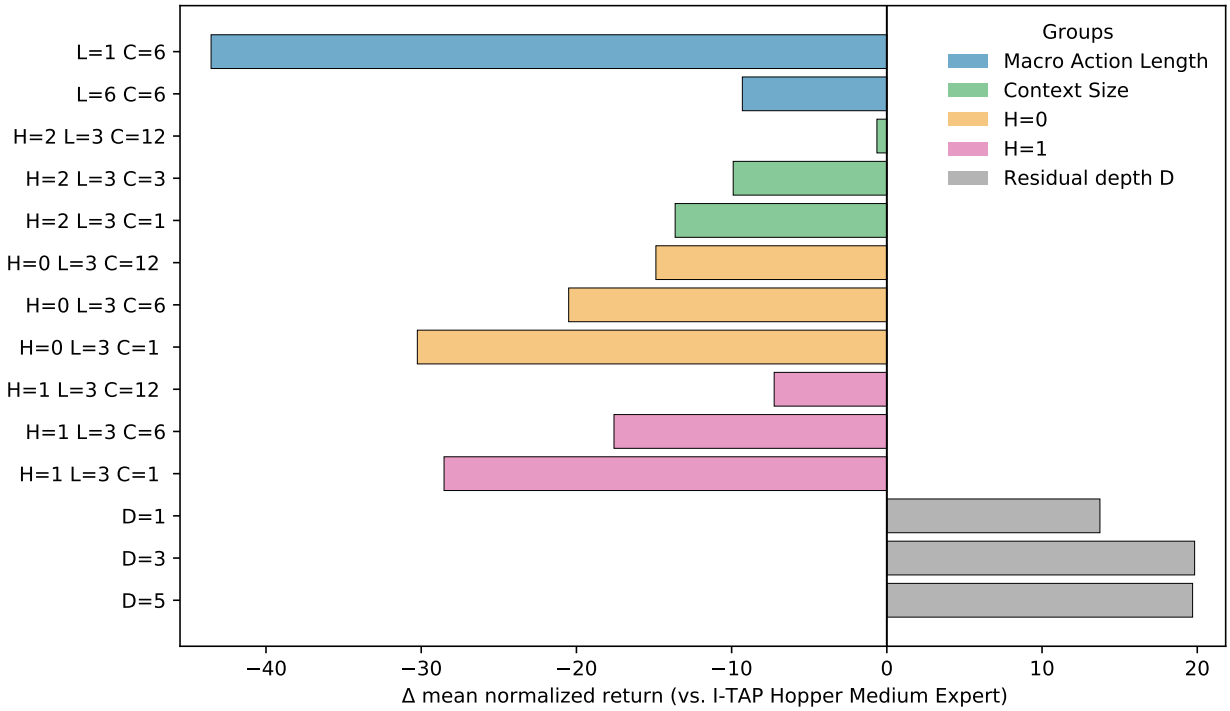


Figure 4: Ablation results across Adroit (expert) and MuJoCo Hopper. We plot Δ scores relative to I-TAP on Hopper medium–expert, which serves as the baseline (zero line).

leverage feedback from the environment to provide more targeted exploration to improve decisions. We use residual depth D=2 for cloned and D=3 for expert datasets; the stronger expert results are consistent with the intuition that coarse-to-fine residual quantization helps retain the granularity needed for high-dimensional continuous control.

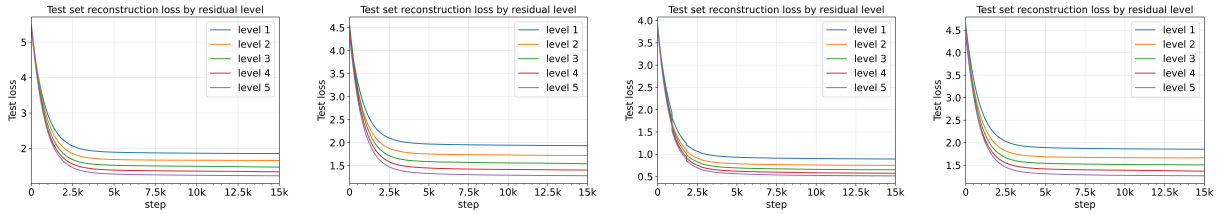
5.2 Ablation Study

We present analyses and ablations of macro action length, context length, planning horizon, and residual depth. Figure 4 summarizes the results from ablation studies conducted across deterministic, moderate-noise, and high-noise Mujoco Hopper control tasks and Adroit (expert) tasks.

Macro Action Length. We vary the macro length L and compare L=1 versus L=6. Increasing to L=6 does *not* noticeably degrade performance, whereas decreasing to L=1 causes a substantial drop. Two factors explain this: (i) with fixed C, the *effective history* observed by the planner scales as $C \times L$ steps, so L=1 shortens the usable history and weakens in-context adaptation; and (ii) shorter macros increase the *branching factor* during search and make the RQ-VAE more prone to *overfitting*, which requires careful hyperparameter control.

Context Size. We ablate context size $C \in \{1, 3, 12\}$ (in latent tokens). The datasets are produced under distinct latent dynamics; thus, context is useful for *inferring the active mode* and for feeding back recent rewards to steer exploration. Increasing context from C=1 (approximately 3 past transitions when L=3) to C=12 (approximately 36 transitions) yields consistent gains. This demonstrates that short, latent contexts already help (few tokens cover many steps), and performance improves as C grows because the model better disambiguates latent parameters and aggregates noisy feedback.

Planning Horizon. We vary the MCTS look-ahead depth H (in latent tokens). With macro length L=3, a depth H corresponds to $H \times L$ planning horizon in the raw action space. Removing planning (H=0) harms performance for all C; the drop is smaller on deterministic tasks where the in-context prior is already strong,



(a) Hammer (b) Door (c) Pen (d) Relocate

Figure 5: Test reconstruction losses across residual levels.

but it remains insufficient under stochasticity. Increasing to H=1 yields consistent gains across settings; at H=2, we observe a pronounced jump, a shorter-context model (C=6) matches a longer-context baseline (C=12). This supports our hypothesis that MCTS could *mitigates uncertainty* due to partial observability and stochastic dynamics by taking expectations over futures: deeper look-ahead can substitute for additional context up to a point before compute and diminishing returns dominate.

Residual Depth We vary the residual quantization depth D for RQ-VAE. Reconstruction error *decreases monotonically* with D (Figure 5), with diminishing returns: the largest drop is from D=1 to D=3. Control performance mirrors this: the task-averaged score rises from 113.40 at D=1 to 119.50 at D=3, and then plateaus (119.37 at D=5). This highlights that increasing residual depth improves reconstruction quality for high-dimensional features and shows that D=3 preserves action granularity for high-DoF control without incurring the computation/optimization overhead of very deep stacks.

5.3 Decision Time vs Context Length

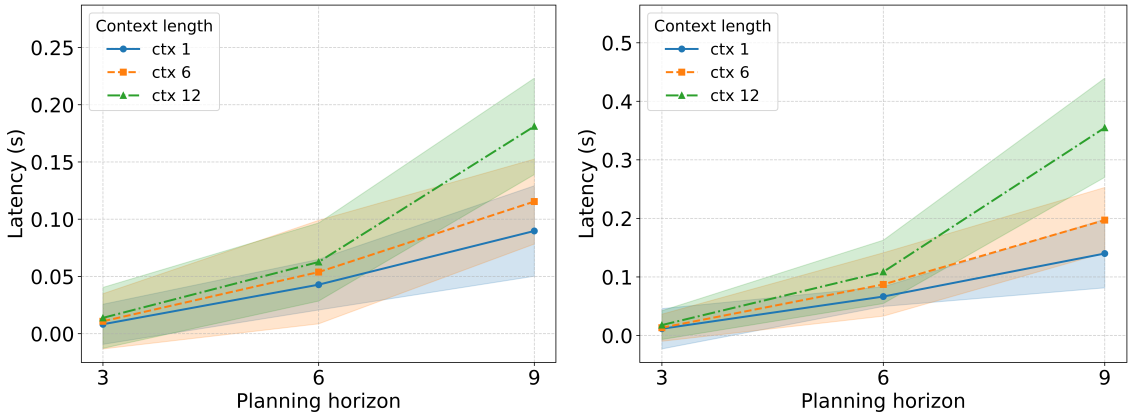


Figure 6: Average decision time vs. context length with 16 action candidates (left) and 32 action candidates (right).

The latency results clearly highlight the trade-off between context size, planning depth, and decision-making efficiency. Our analysis of decision latency versus context length reveals that longer contexts substantially increase the average decision time, particularly as the planning depth increases. For instance, at planning horizon 9, decision latency escalates sharply from approximately 0.09 seconds (context size 1) to 0.35 seconds (context size 12) when the number of action candidates is 32. Similarly, at lower planning depths (1 and 2), longer contexts still cause noticeable latency increases, albeit less dramatically.

Thus, in scenarios characterized by dynamics influenced by unobserved states, our findings underline the necessity of carefully selecting context lengths. It is crucial to encode historical observations effectively, extracting the most informative signals without redundant details that contribute minimal additional predictive power but significantly raise computational latency. In practical applications, particularly those involving

stochasticity and continuous observation-action spaces, balancing sufficient context length to maintain accuracy against the latency imposed by deeper planning becomes a critical design decision. Efficient decision-making, therefore, relies on identifying a context size that captures the essential dynamics without incurring unnecessary computational overhead.

6 Related Works

Reinforcement Learning as Sequence Modeling. Recent advances have reframed reinforcement learning (RL) as a sequence modeling problem, initiated by Decision Transformer (DT), which formulates RL as supervised learning conditioned on desired returns (Chen et al., 2021). Building on these ideas, Algorithm Distillation (Laskin et al., 2023) and Decision-Pretrained Transformer (Lee et al., 2023) leverage Transformers to distill optimal behaviors from historical trajectories, enabling rapid in-context adaptation. More recently, hierarchical variants like In-context Decision Transformer (IDT; Huang et al. (2024)) extend this paradigm by modeling high-level decisions, alleviating computational bottlenecks associated with long context windows. However, as noted by Son et al. (2025), these methods may replicate suboptimal behaviors due to the absence of explicit planning mechanisms, a limitation potentially addressed by model-based planning. Moreover, supervised RL methods typically assume deterministic or near-deterministic datasets, inherently limiting their effectiveness in stochastic environments, where conditioning solely on outcomes can lead to incorrect decisions (Paster et al., 2022). Unlike prior works relying on large contexts in near-deterministic settings, our approach explicitly targets efficient adaptation in dynamic, stochastic environments using a limited context window.

Model-Based Reinforcement Learning. From a model-based perspective (Antonoglou et al., 2022; Schrittwieser et al., 2020), Janner et al. (2021) introduced beam search over a Transformer dynamics model for planning, inspiring subsequent methods that plan in learned latent spaces. In particular, TAP (Jiang et al., 2023) and L-MAP (Luo et al., 2025) employ temporal abstraction by encoding multi-step action segments into discrete codes via state-conditioned VQ-VAEs, then planning over these compact tokens. While beam search with a learned model is effective in largely deterministic settings, L-MAP further adopts MCTS to handle stochastic dynamics and improve robustness. Nevertheless, both methods assume full observability, which can limit performance under state aliasing. Our proposed I-TAP bridges this gap by conditioning planning on recent histories to mitigate partial observability and use MCTS to take expectations over possible futures and deviate from suboptimal priors.

7 Discussion

We presented the *In-Context Latent Temporal Abstraction Planner* (I-TAP), an offline RL approach that discretizes observation–macro-action trajectories and performs planning in a learned latent space. By combining temporal abstraction with in-context conditioning, I-TAP reduces planning complexity while enabling adaptation to latent regime variations through the test-time context window. Residual quantization provides a coarse-to-fine code stack to extend the scalability of the proposed method, and our factorized prior keeps planning-time inference efficient when the model is queried. Across stochastic MuJoCo and high-dimensional Adroit domains, I-TAP achieves competitive performance relative to strong offline baselines. More broadly, the same idea of discretized latent representations plus planning-compatible sequence priors may be useful beyond offline RL, for example in imitation learning settings that require temporally extended actions while retaining fine-grained control.

References

Ioannis Antonoglou, Julian Schrittwieser, Sherjil Ozair, Thomas K. Hubert, and David Silver. Planning in stochastic environments with a learned model. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL `https://openreview.net/forum?id=X6D9bAHhBQ1`.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 15084–15097, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/7f489f642a0ddb10272b5c31057f0663-Abstract.html.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4RL: datasets for deep data-driven reinforcement learning. *CoRR*, abs/2004.07219, 2020. URL https://arxiv.org/abs/2004.07219.

Hiroki Furuta, Yutaka Matsuo, and Shixiang Shane Gu. Generalized decision transformer for offline hindsight information matching. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=CAjxVodl_v.

Sili Huang, Jifeng Hu, Hechang Chen, Lichao Sun, and Bo Yang. In-context decision transformer: Reinforcement learning via hierarchical chain-of-thought. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=jmmji1EU3g.

Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Mohammadamin Barekatain, Simon Schmitt, and David Silver. Learning and planning in complex action spaces. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 4476–4486. PMLR, 2021. URL http://proceedings.mlr.press/v139/hubert21a.html.

Michael Janner, Qiyang Li, and Sergey Levine. Offline reinforcement learning as one big sequence modeling problem. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 1273–1286, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/099fe6b0b444c23836c4a5d07346082b-Abstract.html.

Zhengyao Jiang, Tianjun Zhang, Michael Janner, Yueying Li, Tim Rocktäschel, Edward Grefenstette, and Yuandong Tian. Efficient planning in a compact latent action space. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=cA77NrVEuqn.

Zhengyao Jiang, Yingchen Xu, Nolan Wagener, Yicheng Luo, Michael Janner, Edward Grefenstette, Tim Rocktäschel, and Yuandong Tian. H-GAP: humanoid control with a generalist planner. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=LYG6tBlEX0.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=68n2s9ZJWF8.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/0d2b2061826a5df3221116a5085a6052-Abstract.html.

Michael Laskin, Luyu Wang, Junhyuk Oh, Emilio Parisotto, Stephen Spencer, Richie Steigerwald, DJ Strouse, Steven Stenberg Hansen, Angelos Filos, Ethan Brooks, Maxime Gazeau, Himanshu Sahni, Satinder Singh, and Volodymyr Mnih. In-context reinforcement learning with algorithm distillation. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=hy0a5MMPUv.

Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 11513–11522. IEEE, 2022. doi: 10.1109/CVPR52688.2022.01123. URL https://doi.org/10.1109/CVPR52688.2022.01123.

Jonathan Lee, Annie Xie, Aldo Pacchiano, Yash Chandak, Chelsea Finn, Ofir Nachum, and Emma Brunskill. Supervised pretraining can learn in-context reinforcement learning. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/8644b61a9bc87bf7844750a015feb600-Abstract-Conference.html.

Hao Liu and Pieter Abbeel. Emergent agentic transformer from chain of hindsight experience. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 21362–21374. PMLR, 2023. URL https://proceedings.mlr.press/v202/liu23a.html.

Baiting Luo, Ava Pettet, Aron Laszka, Abhishek Dubey, and Ayan Mukhopadhyay. Scalable decision-making in stochastic environments through learned temporal abstraction. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=pQsllTesiE.

Jianlan Luo, Perry Dong, Jeffrey Wu, Aviral Kumar, Xinyang Geng, and Sergey Levine. Action-quantized offline reinforcement learning for robotic skill learning. In Jie Tan, Marc Toussaint, and Kourosh Darvish (eds.), *Conference on Robot Learning, CoRL 2023, 6-9 November 2023, Atlanta, GA, USA*, volume 229 of *Proceedings of Machine Learning Research*, pp. 1348–1361. PMLR, 2023. URL https://proceedings.mlr.press/v229/luo23a.html.

Amy Mcgovern and Richard Sutton. Macro-actions in reinforcement learning: An empirical analysis. 10 1998.

Keiran Paster, Sheila A. McIlraith, and Jimmy Ba. You can't count on luck: Why decision transformers and rvs fail in stochastic environments. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/fe90657b12193c7b52a3418bdc351807-Abstract-Conference.html.

Wojciech Popko, Matthias Wächter, and Philipp Thomas. Verification of continuous time random walk wind model. volume All Days of *International Ocean and Polar Engineering Conference*, pp. ISOPE–I–16–189, 06 2016.

Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. In Hadas Kress-Gazit, Siddhartha S. Srinivasa, Tom Howard, and Nikolay Atanasov (eds.), *Robotics: Science and Systems XIV, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA, June 26-30, 2018*, 2018. doi: 10.15607/RSS.2018.XIV.049. URL http://www.roboticsproceedings.org/rss14/p49.html.

Marc Rigter, Bruno Lacerda, and Nick Hawes. One risk to rule them all: A risk-sensitive perspective on model-based offline reinforcement learning. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/f49287371916715b9209fa41a275851e-Abstract-Conference.html.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy P. Lillicrap, and David Silver. Mastering atari, go, chess and shogi by planning with a learned model. *Nat.*, 588(7839):604–609, 2020. doi: 10.1038/S41586-020-03051-4. URL https://doi.org/10.1038/s41586-020-03051-4.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. Mastering chess and shogi by self-play with a general reinforcement learning algorithm, 2017. URL https://arxiv.org/abs/1712.01815.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018. doi: 10.1126/science.aar6404. URL https://www.science.org/doi/abs/10.1126/science.aar6404.

Jaehyeon Son, Soochan Lee, and Gunhee Kim. Distilling reinforcement learning algorithms for in-context model-based planning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=BfUugGfBE5.

Richard S. Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1):181–211, 1999. ISSN 0004-3702. doi: https://doi.org/10.1016/S0004-3702(99)00052-1. URL https://www.sciencedirect.com/science/article/pii/S0004370299000521.

Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 6306–6315, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/7a98af17e63a0ac09ce2e96d03992fbc-Abstract.html.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

A Appendix

A.1 Latent Search Tree Construction

Algorithm 1 SampleStack($p_\phi, \mathcal{C}, D, \boldsymbol{\xi}, \boldsymbol{\rho}$)

1: $\mathbf{u}_1 \leftarrow$ scores from $p_\phi(k_1 \mid \mathcal{C})$
2: $k_1 \sim$ TopKTempCat$\big(\mathbf{u}_1, \xi_1, \rho_1\big)$
3: $K \leftarrow (k_1)$
4: **for** $d = 2$ **to** D **do**
5: $\quad\mathbf{u}_d \leftarrow$ scores from $p_\phi(k_d \mid K_{1:d-1}, \mathcal{C})$

$k_d \sim \textsc{TopKTempCat}\big(\mathbf{u}_d,\, \xi_d,\, \rho_d\big)$
$K \leftarrow (K,\, k_d)$ ▷ append
end for
return K

Algorithm 2 Pre-constructing the Latent Search Space (Residual Stack, Macro Tokens)

Require: Current observation o_{t_k}; context $\big((G^{(L)}_{t-cL}, o_{t-cL}, a_{t-cL}), \ldots, (G^{(L)}_{t-1}, o_{t-1}, a_{t-1})\big)$; encoder f_{enc}; decoder f_{dec}; residual-stack model p_ϕ; residual depth D; per-depth temperatures $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_D)$; per-depth top truncation $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_D)$; # coarse samples M; # residual completions per coarse sample J; # lookahead samples N; tree depth H; # kept per node κ_{keep}; # proposals per node B

Ensure: Latent search tree $\mathcal{T}$ with cached promising residual-stack codes

Encode macro-context
$k_{t_{k-1}:t_{k-c},\,1:D} \leftarrow f_{\text{enc}}\big((G^{(L)}_{t-cL}, o_{t-cL}, a_{t-cL}), \ldots, (G^{(L)}_{t-1}, o_{t-1}, a_{t-1})\big)$
Initialize tree $\mathcal{T}$ with root node $s_{t_k} = (o_{t_k},\, k_{t_{k-1}:t_{k-c},\,1:D})$
$\mathcal{C}_k \leftarrow (o_{t_k},\, k_{t_{k-1}:t_{k-c},\,1:D})$
Step 1: sample and score initial macro stacks at index k
/ M coarse draws for depth 1; for each, J residual completions to depth D */*
$\mathbf{u}_{k,1} \leftarrow$ scores from $p_\phi\big(k_{t_k,1} \mid \mathcal{C}_k\big)$
for $i = 1$ **to** M **(parallel) do**
  $k^{(i)}_{t_k,1} \sim \textsc{TopKTempCat}\big(\mathbf{u}_{k,1},\, \xi_1,\, \rho_1\big)$
  for $j = 1$ **to** J **(parallel) do**
    $K^{(i,j)}_{t_k} \leftarrow (k^{(i)}_{t_k,1})$
    for $d = 2$ **to** D **do**
      $\mathbf{u}_{k,d} \leftarrow$ scores from $p_\phi\big(k_{t_k,d} \mid K^{(i,j)}_{t_k,1:d-1},\, \mathcal{C}_k\big)$
      $k^{(i,j)}_{t_k,d} \sim \textsc{TopKTempCat}\big(\mathbf{u}_{k,d},\, \xi_d,\, \rho_d\big)$
      $K^{(i,j)}_{t_k} \leftarrow (K^{(i,j)}_{t_k},\, k^{(i,j)}_{t_k,d})$ ▷ append
    end for
    $z^{(i,j)}_{t_k} \leftarrow \text{Embed}\big(K^{(i,j)}_{t_k}\big)$
    $\widehat{G}^{(L)}_{t_k}\big(K^{(i,j)}_{t_k}\big) \leftarrow$ current-step head from f_{dec}
    for $n = 1$ **to** N **(parallel) do**
      $\mathcal{C}' \leftarrow (o_{t_k},\, K^{(i,j)}_{t_k},\, k_{t_{k-1}:t_{k-c},\,1:D})$
      $K^{(i,j,n)}_{t_{k+1}} \leftarrow \text{SampleStack}(p_\phi,\, \mathcal{C}',\, D,\, \boldsymbol{\xi},\, \boldsymbol{\rho})$
      $z^{(i,j,n)}_{t_{k+1}} \leftarrow \text{Embed}\big(K^{(i,j,n)}_{t_{k+1}}\big)$
      $\hat{y}^{(i,j,n)}_{t_{k+1}} \leftarrow f_{\text{dec}}\big(z^{(i,j)}_{t_k},\, z^{(i,j,n)}_{t_{k+1}},\, o_{t_k},\, k_{t_{k-1}:t_{k-c},\,1:D}\big)$
    end for
    $\text{score}\big(K^{(i,j)}_{t_k}\big) \leftarrow \frac{1}{N}\sum_{n=1}^{N}\Big(\widehat{G}^{(L)}_{t_k}\big(K^{(i,j)}_{t_k}\big) + \big[\hat{y}^{(i,j,n)}_{t_{k+1}}\big]_{\text{rtg}}\Big)$
    $\bar{y}^{(i,j)}_{t_{k+1}} \leftarrow \frac{1}{N}\sum_{n=1}^{N}\hat{y}^{(i,j,n)}_{t_{k+1}}; \quad \hat{o}^{(i,j)}_{t_{k+1}} \leftarrow \text{Obs}\big(\bar{y}^{(i,j)}_{t_{k+1}}\big)$
  end for
end for
Select top-κ_{keep} stacks $\{K^{(i,j)}_{t_k}\}$ by score; for each, attach child node $\big(\hat{o}^{(i,j)}_{t_{k+1}},\, K^{(i,j)}_{t_k}\big)$ under the root in $\mathcal{T}$
Step 2: recursive latent-tree expansion over macro indices
for $h = 2$ **to** H **do**
  Let $\mathcal{N}_{h-1}$ be the nodes at depth $h-1$ of $\mathcal{T}$
  for each node $(\hat{o}, K^c) \in \mathcal{N}_{h-1}$ **(parallel) do**
    $\mathcal{C} \leftarrow (\hat{o},\, K^c,\, k_{t_{k-1}:t_{k-c},\,1:D})$
    for $b = 1$ **to** B **(parallel) do**
      $K^{(b)}_{t_{k+h-1}} \leftarrow \text{SampleStack}(p_\phi,\, \mathcal{C},\, D,\, \boldsymbol{\xi},\, \boldsymbol{\rho})$
      $z^{(b)}_{t_{k+h-1}} \leftarrow \text{Embed}\big(K^{(b)}_{t_{k+h-1}}\big)$
      $\widehat{G}^{(L)}_{t_{k+h-1}}\big(K^{(b)}_{t_{k+h-1}}\big) \leftarrow$ current-step head from f_{dec}
      for $n = 1$ **to** N **(parallel) do**
        $\mathcal{C}^\star \leftarrow (\hat{o},\, K^{(b)}_{t_{k+h-1}},\, k_{t_{k-1}:t_{k-c},\,1:D})$
        $K^{(b,n)}_{t_{k+h}} \leftarrow \text{SampleStack}(p_\phi,\, \mathcal{C}^\star,\, D,\, \boldsymbol{\xi},\, \boldsymbol{\rho})$
        $z^{(b,n)}_{t_{k+h}} \leftarrow \text{Embed}\big(K^{(b,n)}_{t_{k+h}}\big)$
        $\hat{y}^{(b,n)}_{t_{k+h}} \leftarrow f_{\text{dec}}\big(z^{(b)}_{t_{k+h-1}},\, z^{(b,n)}_{t_{k+h}},\, \hat{o},\, k_{t_{k-1}:t_{k-c},\,1:D}\big)$

```
            end for
            score(K^(b)_{t_{k+h-1}}) ← (1/N) Σ_{n=1}^{N} ( Ĝ^(L)_{t_{k+h-1}}(K^(b)_{t_{k+h-1}}) + [ŷ^(b,n)_{t_{k+h}}]_rtg )
            ȳ^(b)_{t_{k+h}} ← (1/N) Σ_{n=1}^{N} ŷ^(b,n)_{t_{k+h}};   ô^(b)_{t_{k+h}} ← Obs(ȳ^(b)_{t_{k+h}})
        end for
        Select top-κ_keep from {K^(b)_{t_{k+h-1}}}_{b=1}^{B} by score and attach as children (ô^(b)_{t_{k+h}}, K^(b)_{t_{k+h-1}}) under (ô, K^c) in T
    end for
end for
return T
```

A.2 Experiment Details

A.2.1 Implementation Details

The hyperparameter settings for I-TAP are listed in Table 4. For baselines, we follow the implementations and recommended hyperparameters from the original works, including TAP (Jiang et al., 2023), L-MAP (Luo et al., 2025), CQL (Kumar et al., 2020), IQL Kostrikov et al. (2022), DT (Chen et al., 2021), and 1R2R (Rigter et al., 2023). For the Stochastic MuJoCo benchmarks, the results of TAP, 1R2R, CQL, and IQL are taken from the L-MAP paper (Luo et al., 2025), which evaluates these methods under the same stochastic-domain protocol. For deterministic domains, the CQL and IQL results are taken from their respective papers (Kumar et al., 2020; Kostrikov et al., 2022). Each run of I-TAP takes approximately 6 hours on one NVIDIA RTX 5090 GPU with an Intel(R) Core(TM) i9-14900KS CPU.

A.2.2 Domains

Stochastic MuJoCo. The Stochastic MuJoCo tasks introduced by Rigter et al. (2023) apply incremental perturbation forces following a uniform random walk (Popko et al., 2016). At each timestep, the perturbation force f_t is updated as:

$$f_{t+1} = f_t + \Delta f, \quad \Delta f \sim \text{Uniform}\left(-0.1 \cdot f_{\text{MAX}},\ 0.1 \cdot f_{\text{MAX}}\right), \tag{1}$$

with the total perturbation clipped to remain within $[-f_{\text{MAX}}, f_{\text{MAX}}]$. This model introduces persistent, incremental perturbations, representing a baseline scenario. For both Hopper and Walker2D environments, the perturbation force f_t is applied horizontally along the x-axis to simulate external disturbances, such as wind gusts. The maximum perturbation magnitude f_{MAX} specifically for the Hopper moderate perturbation level is 2.5 Newtons, high perturbation level is 5 Newtons, Walker2D moderate perturbation level is 7 Newtons, high perturbation level is 12 Newtons.

Adroit as a POMDP. Many Adroit manipulation tasks expose *privileged* channels that directly encode the goal state and progress (e.g., target poses, object–target deltas, insertion depth). To evaluate agents under partial observability, we define a POMDP variant that preserves the original dynamics and reward but *ablates* these channels at evaluation time. Concretely, let the environment produce an observation $o_t \in \mathbb{R}^d$ under the standard `v1` layout. We introduce a fixed binary mask $m \in \{0,1\}^d$ (zeros at privileged indices) and set

$$\tilde{o}_t \;=\; m \odot o_t, \qquad O(\tilde{o}_t \mid s_t) \;=\; \delta\big(\tilde{o}_t - m \odot o_t\big),$$

yielding a POMDP in which progress toward the goal must be *inferred* from history rather than read off directly.

Masking regimes. Indices refer to 0-based positions in the default `v1` observation vector.

- **Pen** (d=45): $\mathcal{I} = \{36, 37, 39, 40, 42, 43\}$.
- **Relocate** (d=39): $\mathcal{I} = \{30, \ldots, 38\}$.
- **Door** (d=39): $\mathcal{I} = \{27, 28, 32, \ldots, 38\}$.
- **Hammer** (d=46): $\mathcal{I} = \{43, 44, 45\}$.

Table 4: List of Hyper-parameters

Environment	**Hyper-parameter**	**Value**
All	learning rate	1×10^{-4}
All	batch size	512
All	dropout probability	0.1
All	number of attention heads	4
All	macro action length L	3
All	embedding size (latent code)	512
All	c_1	1.25
All	c_2	19652
All	α_{tail}	1
All	α_{ctx}	0.1
All	β_{ps}	1
MuJoco	context length c/training sequence length	6/24
MuJoco	discount factor	0.99
MuJoco	number of Transformer layers	4
MuJoco	feature vector size	512
MuJoco	codebook size	512
MuJoco	initial number of policy samples M	16
MuJoco	number of transition samples N	4
MuJoco	number of policy samples B	4
MuJoco	number of MCTS iterations	100
Mujoco	κ_{keep}	50%
MuJoco	temperature	2
MuJoco	Residual Depth	1
MuJoco	primitive planning horizon	9
Adroit	context length c/training sequence length	6/24
Adroit	discount factor	0.99
Adroit	number Transformer layers	4
Adroit	feature vector size	256
Adroit	codebook size	512
Adroit	initial number of policy samples M	16
Adroit	number of transition samples N	4
Adroit	number of policy samples B	4
Adroit	number of MCTS iterations	100
Adroit	Residual Depth	2,3
Adroit	κ_{keep}	10%
Adroit	J	4
Adroit	temperature	1
Adroit	primitive planning horizon	9

This construction leaves proprioception and contact signals intact while removing privileged goal vectors and progress proxies, thereby converting the original fully observable tasks into history-dependent POMDPs that better reflect realistic sensing and require temporal credit assignment and state estimation.