LARGE LANGUAGE MODELS FOR SEQUENTIAL DECISION-MAKING: IMPROVING IN-CONTEXT LEARNING VIA SUPERVISED FINE-TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) have shown remarkable in-context learning (ICL) capabilities, yet their potential for sequential decision-making remains underexplored. In this paper, we study the ICL capabilities of LLMs in sequential decision-making settings, including Markov Decision Processes (MDPs), Partially Observable MDPs (POMDPs), and Ambiguous POMDPs (APOMDPs). We fine-tune pretrained LLMs to perform few-shot decision-making directly from offline, oracle-labeled trajectories. Our framework enables flexible imitation of policies through supervised fine-tuning (SFT). Theoretically, we focus on linear MDPs and interpret a fine-tuned attention layer as implicitly estimating optimal Q-functions from in-context data. Building on this interpretation, we derive an end-to-end suboptimality bound for the induced policy that separates the in-context estimation error from the training-length bias. Empirically, across synthetic MDP, POMDP, and APOMDP settings, we find that fine-tuned LLMs achieve substantially smaller optimality gaps than in-context-only and random baselines, with especially large gains in longer-horizon, partially observed, and model-ambiguous environments. Together, these results show that supervised fine-tuning provides an effective route to endowing pretrained LLMs with sequential decision-making capabilities from offline data, which is an important advantage in domains such as healthcare where offline data are abundant.

1 INTRODUCTION

Sequential decision-making under uncertainty is central to many high-stakes domains, from clinical treatment planning to resource management. In these settings, an agent must choose a sequence of actions over time to maximize cumulative rewards, using only observational data collected under some historical policy. Dynamic Treatment Regimes (DTRs) provide a principled framework for discovering effective policies from such data (Robins, 1986; Murphy, 2003). However, standard DTR approaches rely on the strong assumption that all confounders are observable or that their effects can be ignored (sequential ignorability), which is often violated in practice. To address this challenge, Ambiguous Dynamic Treatment Regimes (ADTRs) account for unobservable confounders by evaluating policies against a set of plausible data-generating models (model ambiguity) rather than a single one (Saghafian, 2018; 2024; 2025). Furthermore, previous studies such as Saghafian (2024) connect ADTRs to APOMDPs and develop reinforcement learning (RL) approaches to find the best sequence of actions. Compared with MDPs or POMDPs, APOMDPs provide a more realistic framework for learning sequential decision-making from observational data in the presence of unobservable confounders by directly considering the underlying model ambiguity.

Large language models have shown remarkable in-context learning capabilities. Given a small set of demonstrations as the input prompt, a pretrained LLM can make predictions on new instances without updating its parameters (Brown et al., 2020). This ability has been studied in supervised learning (Garg et al., 2022; Zhang et al., 2024a) and, more recently, in sequential decision-making settings (Chen et al., 2021; Laskin et al., 2022; Lee et al., 2023). However, prior work on in-context learning for sequential decision-making mainly trains transformer architectures from scratch, focuses on fully observable MDPs, and does not consider the real-world challenges caused by partial observability or model ambiguity.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

In this study, we adapt supervised fine-tuning of pretrained LLMs to sequential decision-making in MDPs, POMDPs, and APOMDPs. Rather than training a transformer from scratch, we fine-tune an open-source pretrained LLM so that it can first read a few demonstration trajectories and then act in new, unseen tasks without any further parameter updates. Building on pretrained LLMs leverages the broad representational capacity acquired during pretraining, which may improve generalization across diverse tasks. Starting from pretrained LLMs also provides an avenue for integrating diverse inputs and human intuition inclusion in real-world sequential decision-making applications, augmenting the capabilities of human-AI centaurs (Saghafian, 2025; Saghafian & Idan, 2024). Our main contributions in this work are as follows.

- **SFT framework for in-context sequential decision-making.** We fine-tune a pretrained LLM on offline trajectories so that it can make sequential decisions in new tasks given only a few demonstrations at test time. The framework uses parameter-efficient adapters and trains on serialized trajectories. At test time, this approach allows the model to read few-shot demonstrations and act in new tasks without any further parameter updates.
- **Offline data use with oracle supervision.** Our framework leverages offline datasets along with an optimal (or near-optimal) action oracle to synthesize high-quality trajectories for fine-tuning. This practice is particularly appealing in domains such as healthcare, where experimentation is costly (or unethical) and model-based or expert-derived supervision can often be constructed from logged observational data.
- **Theoretical analysis in linear MDPs.** Adopting the in-context linear-regression framework of Zhang et al. (2024a), we interpret a trained linear self-attention layer as implicitly estimating the parameters of the optimal Q-function for a new task. Our theoretical contribution (a) makes use of this prediction-level interpretation to create an end-to-end suboptimality bound for the induced policy and (b) decomposes the bound into an in-context estimation error (decreasing in the number of support trajectories) and a training-length bias (decreasing in the number of training trajectories).
- **Empirical evaluation across MDPs, POMDPs, and APOMDPs.** We show that fine-tuned LLMs achieve low optimality gaps and consistently outperform both random policies and ICL-only baselines. Specifically, we find that fine-tuning roughly halves the optimality gap in long-horizon MDPs. In POMDPs and APOMDPs, fine-tuned LLMs achieve large gains even under partial observability and model ambiguity, suggesting robust performance for our approach across different settings.

2 Related work

In-context prediction. Existing research has investigated ICL in classification and regression (Garg et al., 2022; Zhang et al., 2024a; Mirchandani et al., 2023; Zhao et al., 2024). Our analysis builds on the linear-regression guarantees of Zhang et al. (2024a) to study policy suboptimality.

LLMs as decision-making agents. Off-the-shelf LLMs can struggle with in-context exploration (Krishnamurthy et al., 2024). EVOLvE (Nie et al., 2025) addresses this through demonstrations and SFT on oracle-generated bandit trajectories. Park et al. (2026) iteratively fine-tune on low-regret model rollouts. Chen et al. (2026) compare SFT and RL for bandit decision-making, while Schmied et al. (2026) study RL fine-tuning and exploration failures. Our study builds on trajectory-based policy distillation, using fixed offline trajectories from MDP, POMDP, and APOMDP solvers.

In-context decision-making. Decision Transformer was among the first to cast RL as sequence modeling, training a transformer to predict actions conditioned on past states, actions, and a target return (Chen et al., 2021). Algorithm Distillation instead trains transformers to imitate the behavior of RL algorithms from their learning histories (Laskin et al., 2022). Decision-Pretrained Transformer (DPT) trains a transformer to imitate an optimal action oracle given in-context data (Lee et al., 2023). Lin et al. (2024) provide guarantees for supervised pretraining to imitate RL algorithms, while Wang et al. (2025) study in-context temporal-difference policy evaluation. Raparthy et al. (2024) investigate training-data properties that enable generalization to new decision-making tasks.

Our study extends this literature in two aspects. First, we examine policy distillation across planning horizons, observation quality, and model ambiguity using a common pretrained-LLM SFT approach.

APOMDPs allow us to vary both the set of plausible models and the decision maker's ambiguity attitude (Saghafian, 2018), adding a decision criterion beyond observation noise. Second, our controlled linear-attention analysis translates Q-prediction error into a policy-gap bound separating in-context estimation error from training-length bias. We provide more discussion of related studies in Appendix A.

3 Problem setup and LLMs

We first present a general framework for sequential decision-making that covers MDPs, POMDPs, and APOMDPs as special cases. We then describe two approaches to preparing LLMs for sequential decision-making problems.

3.1 Sequential decision-making

We consider a setup in which an agent makes a sequence of decisions over a finite horizon T_τ for each task τ drawn from a distribution $\mathcal{T}$. A task τ may vary across draws in its transition dynamics, reward function, and state, action, and reward spaces. For a fixed task τ, let $\mathcal{X}_\tau$, $\mathcal{A}_\tau$, and $\mathcal{R}_\tau$ denote the observation, action, and reward spaces, respectively.

At the beginning of each period $t \in \{1, \dots, T_\tau\}$, the agent first observes $X_t \in \mathcal{X}_\tau$, then chooses an action $a_t \in \mathcal{A}_\tau$, after which a reward $r_t \in \mathcal{R}_\tau$ is realized. The observable history before action a_t is $H_t \triangleq (X_1, a_1, r_1, \dots, X_{t-1}, a_{t-1}, r_{t-1}, X_t)$. A policy is a sequence $\pi_\tau = (\pi_{\tau,t})_{t=1}^{T_\tau}$, where $\pi_{\tau,t}(\cdot \mid H_t) \in \Delta(\mathcal{A}_\tau)$ maps the observable history to a distribution over actions and we denote by $\Delta(\mathcal{A}_\tau)$ the probability simplex over the action space $\mathcal{A}_\tau$. Given a discount factor $\gamma \in (0, 1]$, the performance is $J_\tau(\pi_\tau) \triangleq \mathbb{E}[\sum_{t=1}^{T_\tau} \gamma^{t-1} r_t]$. The goal is to find a policy that maximizes this performance. In what follows, we suppress the subscript τ when the task is clear from context.

3.2 Special cases

We discuss three special cases under the assumption that transition dynamics are Markovian. We provide the formal definitions and detailed notation for these cases in Appendix B.

MDP [full observability]. In an MDP task $\mathcal{E}_\tau^{\mathrm{MDP}} = \langle \mathcal{S}_\tau, \mathcal{A}_\tau, P_\tau, R_\tau, \rho_\tau, T_\tau, \gamma \rangle$, the state $S_t \in \mathcal{S}_\tau$ is fully observable (i.e., $X_t = S_t$). We denote by $P_\tau(\cdot \mid s, a)$ the state-transition kernel, $R_\tau(s, a)$ the one-step reward, and ρ_τ the initial-state distribution. The Markov property allows restriction to policies $\pi_t(a \mid S_t)$.

POMDP [partial observability]. When we observe only $O_t \in \mathcal{O}_\tau$ generated from a latent state $S_t \in \mathcal{S}_\tau$ via an observation kernel $Q_\tau(o \mid s, a)$, we set $X_t = O_t$. We maintain a belief vector $b_t \in \Delta(\mathcal{S}_\tau)$ updated via a Bayes operator B_τ. The belief vector serves as a sufficient statistic for decision-making, which means we can restrict attention to belief-based policies $\pi_t(a \mid b_t)$.

APOMDP [model ambiguity]. Using a single transition or observation kernel is often unreliable for real-world data. APOMDPs address this problem by equipping the task with an ambiguity set $\mathcal{M}_\tau$ of plausible models m, each specifying kernels P_τ^m and Q_τ^m:

$$\mathcal{E}_\tau^{\mathrm{APOMDP}} = \big\langle \mathcal{S}_\tau,\ \mathcal{A}_\tau,\ \mathcal{O}_\tau,\ \mathcal{M}_\tau,\ P_\tau^m,\ Q_\tau^m,\ R_\tau,\ \rho_\tau,\ T_\tau,\ \gamma,\ \alpha \big\rangle,$$

where α tunes pessimism (Saghafian, 2018; 2024). One can derive an optimal policy that hedges against model ambiguity by solving the Bellman equation $V_t(b_t, \alpha) = \max_{a \in \mathcal{A}_\tau} \{U_t(b_t, \alpha, a)\}$, where

$$U_t(b_t, \alpha, a) = b_t r_t + \alpha\gamma \min_{m \in \mathcal{M}_\tau} \{\mathcal{H}_{t+1}(b_t, \alpha, a, m)\} + (1 - \alpha)\gamma \max_{m \in \mathcal{M}_\tau} \{\mathcal{H}_{t+1}(b_t, \alpha, a, m)\},$$

and $\mathcal{H}_{t+1}(b, \alpha, a, m) \triangleq \sum_{o \in \mathcal{O}_\tau} \mathbb{P}(o|b, a, m) V_{t+1}(B_\tau(b, a, o, m), \alpha)$ (Saghafian, 2018). Setting $\alpha = 1$ yields the pessimistic maximin criterion and $\alpha = 0$ the optimistic maximax (Saghafian, 2018; 2024). APOMDPs generalize both MDPs and POMDPs by incorporating model ambiguity. When $\mathcal{M}_\tau$ is a singleton, MDPs arise under full observability and POMDPs arise under partial observability.

3.3 LLMs FOR SEQUENTIAL DECISION-MAKING

We formulate sequential decision-making as an optimization problem in the settings mentioned above. An alternative framework is to recast the problem as a sequence-generation problem. Instead of directly optimizing the performance, an LLM learns a predictor of actions given the history H_t. Training proceeds over trajectories from diverse tasks τ, and at test time the LLM generates actions in new tasks. This framework connects RL-style optimization with generative modeling, in which we view the policy as an inference mechanism that generates optimal actions conditioned on observations. Figure 1 illustrates the overall framework. In what follows, we consider two approaches, in-context learning and supervised fine-tuning, to prepare LLMs for sequential decision-making.

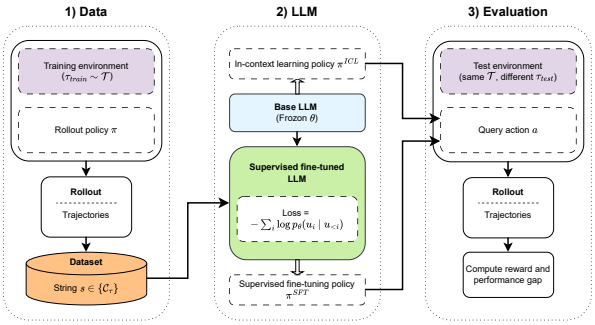

Figure 1: Overview of data construction, policy generation, and performance evaluation.

In-context learning. ICL treats an LLM as a policy generator that adapts at inference time by conditioning on demonstrations in the prompt. We first draw a task τ_{test} from $\mathcal{T}$. We then compute its optimal policy and roll out multiple few-shot support trajectories $\mathcal{D}_\tau$. Next, we serialize those trajectories into a textual context $\text{ctx}_\tau = \mathsf{E}(\mathcal{D}_\tau)$ via a deterministic encoder E. Finally, the LLM receives (H_t, ctx_τ) at the beginning of each period t and generates an action $a_t \sim \pi^{\text{ICL}}(\cdot \mid H_t, \text{ctx}_\tau)$, where we denote by $\pi^{\text{ICL}} \in \Delta(\mathcal{A}_\tau)$ the in-context learning policy. In particular, we restrict outputs to action tokens and select the one with the highest probability.

Supervised fine-tuning. Fine-tuning adapts the pretrained parameters of the LLM to the target task distribution. We adopt parameter-efficient quantized low-rank adapters (QLoRA) (Dettmers et al., 2023), which enable learning a low-rank update for each linear module while keeping the base weights frozen. For each training task τ_{train}, we roll out multiple demonstration trajectories $\mathcal{C}_\tau$ under the optimal policy. We then serialize each trajectory into a string s. The training set comprises trajectories from multiple tasks drawn from $\mathcal{T}$. After tokenization $s \mapsto (u_1, \ldots, u_L)$, SFT proceeds by minimizing the following loss function to obtain the optimal parameters θ^*:

$$\mathcal{L}_{\text{SFT}}(\theta) \;=\; \mathbb{E}_{s \in \{\mathcal{C}_\tau\}} \left[-\sum_{i=1}^{L} \log p_\theta\big(u_i \,\big|\, u_{<i}\big) \right]. \tag{1}$$

This loss function matches the inference-time format. That is, the LLM learns to read few-shot demonstrations and continue the query trajectory in the same schema. We provide more details of fine-tuning in Appendix C. At evaluation, we use the same few-shot support structure as in the ICL setting, but now the policy is $\pi^{\text{SFT}}(\cdot \mid H_t, \text{ctx}_\tau)$. We provide more detailed pseudocode in Appendix D.

4 THEORETICAL ANALYSIS

We develop a theoretical framework to understand why SFT improves in-context sequential decision-making. For tractability, our theoretical analysis considers a single linear self-attention (LSA) layer trained by gradient flow to predict optimal Q-values in linear MDPs. We build on the in-context linear-regression analysis of Zhang et al. (2024a), which enables us to translate prediction-level

guarantees into policy-level statements. We provide an end-to-end bound on the suboptimality of the policy induced by the LSA predictions, with an associated sample-complexity result (deferred to the appendix) relating the number of few-shot support trajectories and the training trajectory length to performance.

4.1 Setup: In-context Q-value prediction in linear MDPs

We consider a linear MDP task $\tau = \langle \mathcal{S}, \mathcal{A}, P, R, \rho, \gamma, \phi \rangle$ with known feature map $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$. For a fixed task τ, the optimal action-value function is assumed to be linear:

$$Q^*_\tau(s,a) = \langle \phi(s,a), w^*_\tau \rangle, \qquad w^*_\tau \in \mathbb{R}^d. \tag{2}$$

For finite-horizon tasks, the time index can be absorbed into the feature map, e.g., by replacing $\phi(s,a)$ with $\phi_t(s,a)$ or $\phi(s,a,t)$. We suppress this notationally and write $\phi(s,a)$ throughout. We assume $w^*_\tau \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}_d,\, I_d)$ after a standard rescaling of the features, where $\mathbf{0}_d \in \mathbb{R}^d$ denotes the zero vector and $I_d \in \mathbb{R}^{d\times d}$ the identity matrix.

Fixed-covariance reduction to in-context linear regression. Now let us fix a task τ. We denote by μ the state-action distribution used to form the in-context examples, and define the feature covariance $\Lambda \triangleq \mathbb{E}_{(s,a)\sim\mu}\big[\phi(s,a)\phi(s,a)^\top\big] \in \mathbb{R}^{d\times d}$. Throughout the formal analysis we assume $\Lambda \succ 0$ is fixed across meta-training and test prompts, matching the fixed-covariance regime of Zhang et al. (2024a). This holds, e.g., when the support trajectories come from a task-independent exploratory policy, or when the task distribution is restricted so that all tasks induce the same feature covariance. If Λ varies, Zhang et al. (2024a) show that a single LSA layer can become biased under such covariate shifts. We therefore treat fixed Λ as an explicit modeling assumption.

Let $x_i \triangleq \phi(s_i,a_i)$ and $y_i \triangleq Q^*_\tau(s_i,a_i) = \langle x_i, w^*_\tau \rangle$. Following Zhang et al. (2024a), we approximate the feature distribution by the matched Gaussian model $x_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}_d, \Lambda)$. The K_{test} few-shot support trajectories each have length T. After the regression reduction, this gives $M \triangleq K_{\text{test}}T$ in-context pairs $(x_i, y_i)_{i=1}^M$, together with a query feature $x_{\text{q}} = \phi(s_{\text{q}}, a_{\text{q}})$. Training trajectories contain $N \triangleq KT$ pairs from K training trajectories.

LSA model and population training. Given $P = (x_1, y_1, \ldots, x_M, y_M, x_{\text{q}})$, we define the embedding matrix $E = \left(\begin{smallmatrix} x_1 & \cdots & x_M & x_{\text{q}} \\ y_1 & \cdots & y_M & 0 \end{smallmatrix}\right) \in \mathbb{R}^{(d+1)\times(M+1)}$. The LSA layer with parameters $\theta = (W_{KQ}, W_{PV})$ computes $f_{\text{LSA}}(E;\theta) = E + W_{PV}E\,E^\top W_{KQ}E/M$. The predicted Q-value $\widehat{Q}(x_{\text{q}};\theta)$ is the bottom-right entry of $f_{\text{LSA}}(E;\theta)$. The LSA layer is trained by gradient flow on the population mean-squared error

$$L(\theta) = \frac{1}{2}\mathbb{E}_{w,x_i,x_{\text{q}}}\left[\left(\widehat{Q}(x_{\text{q}};\theta) - \langle w, x_{\text{q}}\rangle\right)^2\right], \tag{3}$$

where $w \sim \mathcal{N}(\mathbf{0}_d, I_d)$ and $x_i, x_{\text{q}} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}_d, \Lambda)$. All formal results below are stated for this population objective. With finitely many training tasks, additional statistical and optimization errors can be added to the prediction error and we do not assign a specific rate to that finite-sample term here.

4.2 In-context Q-prediction error

The first ingredient is a per-query Q-prediction guarantee for the trained LSA layer. Define

$$\Gamma \triangleq \left(1 + \frac{1}{N}\right)\Lambda + \frac{\operatorname{tr}(\Lambda)}{N} I_d. \tag{4}$$

The following lemma is a direct restatement of Theorems 4.1–4.2 and Corollary 4.3 of Zhang et al. (2024a) in our linear MDP notation.

Lemma 1 (In-context Q-prediction error)**.** *Suppose the initialization satisfies Assumption 3.3 of Zhang et al. (2024a), with initialization scale small enough for their convergence theorem. Gradient flow on* (3) *converges to a global minimizer θ^*. At this minimizer, for any $P = (x_1, y_1, \ldots, x_M, y_M, x_{\text{q}})$*

drawn from an unseen task with $w^* \sim \mathcal{N}(\mathbf{0}_d, I_d)$, $x_i, x_{\mathrm{q}} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}_d, \Lambda)$, *and* $y_i = \langle w^*, x_i \rangle$, *the LSA prediction is*

$$\widehat{Q}(x_{\mathrm{q}} \mid P) = x_{\mathrm{q}}^\top \Gamma^{-1} \left(\frac{1}{M} \sum_{i=1}^{M} y_i x_i \right). \tag{5}$$

Moreover, in the noiseless realizable case,

$$\varepsilon_Q(M, N) \triangleq \mathbb{E}_{w^*, x_i, x_{\mathrm{q}}} \Big[\big(\widehat{Q}(x_{\mathrm{q}} \mid P) - \langle w^*, x_{\mathrm{q}} \rangle \big)^2 \Big] \leq \frac{(d+1)\operatorname{tr}(\Lambda)}{M} + \frac{(1 + 2d + d^2\kappa)\operatorname{tr}(\Lambda)}{N^2}, \tag{6}$$

where κ *is the condition number of* Λ.

Equation (5) shows that the trained LSA layer implicitly computes a covariance-corrected in-context estimator. The factor Γ^{-1} acts as a shrinkage version of Λ^{-1} whose bias depends on the length $N = KT$. As $N \to \infty$, $\Gamma^{-1} \to \Lambda^{-1}$, and the estimator approaches the ordinary least-squares form. The two terms in (6) have distinct origins: the $1/M$ term is the in-context estimation error on an unseen task, while the $1/N^2$ term is the training-length bias.

4.3 End-to-end suboptimality of the induced policy

Lemma 1 bounds the error of one Q-value query under the matched Gaussian query model. Our experiments evaluate the policy obtained by acting with respect to the predicted Q-values along a full trajectory. To lift the prediction guarantee to a policy guarantee, we need an assumption that the Q-prediction error remains controlled on the query points encountered by the induced policy.

On-policy query calibration. Let $\widehat{\pi}$ be the policy induced by $\widehat{Q}$: $\widehat{\pi}_t(s) \in \arg\max_{a \in \mathcal{A}} \widehat{Q}_t(s, a)$. We assume a constant $C_{\mathrm{on}} \geq 1$ such that, for every time $t \leq T$,

$$\max_{\pi'_t \in \{\pi^*_t, \widehat{\pi}_t\}} \mathbb{E} \left[\Big(\widehat{Q}_t(s_t, \pi'_t(s_t)) - Q^*_t(s_t, \pi'_t(s_t)) \Big)^2 \right] \leq C_{\mathrm{on}} \varepsilon_Q(M, N), \tag{7}$$

where the expectation is over the support trajectory, the unseen task, and the trajectory distribution $s_t \sim d_t^{\widehat{\pi}}$. This assumption is a direct on-policy transfer condition: it rules out the covariate-shift failure modes identified by Zhang et al. (2024a). In the ideal matched-query setting, one may take $C_{\mathrm{on}} = 1$; otherwise C_{on} measures the mismatch between the Gaussian query model in Lemma 1 and the state-action queries encountered during rollout.

Proposition 1 (End-to-end optimality gap)**.** *Let* τ_{new} *be an unseen linear MDP drawn from* $\mathcal{T}$*, with horizon* T*, discount factor* $\gamma \in (0, 1]$*, and optimal Q-functions linear in the features as in* (2)*. Let* $\widehat{Q}$ *be the LSA prediction obtained from a length-*M *support trajectory, and let* $\widehat{\pi}$ *be the policy induced by* $\widehat{Q}$*. Under the conditions of Lemma 1 and the on-policy calibration condition* (7)*,*

$$\mathbb{E}_{\tau_{\mathrm{new}}} \left[J_{\tau_{\mathrm{new}}}(\pi^*) - J_{\tau_{\mathrm{new}}}(\widehat{\pi}) \right] \leq C_{T,\gamma} \sqrt{C_{\mathrm{on}} \varepsilon_Q(M, N)}, \tag{8}$$

where $C_{T,\gamma} \triangleq 2(1 - \gamma^T)/(1 - \gamma)$ *and* $C_{T,1} \triangleq 2T$*. Substituting* (6) *gives*

$$\mathbb{E} \left[J_{\tau_{\mathrm{new}}}(\pi^*) - J_{\tau_{\mathrm{new}}}(\widehat{\pi}) \right] \leq C_{T,\gamma} \sqrt{C_{\mathrm{on}}} \left[\frac{(d+1)\operatorname{tr}(\Lambda)}{M} + \frac{(1 + 2d + d^2\kappa)\operatorname{tr}(\Lambda)}{N^2} \right]^{1/2}. \tag{9}$$

Implications. Substitute $M = K_{\mathrm{test}} T$ and $N = KT$. In the nearly undiscounted regime $\gamma \to 1$, (9) implies

$$\left(\mathbb{E}[J(\pi^*) - J(\widehat{\pi})] \right)^2 \lesssim C_{\mathrm{on}} \left[\frac{4T(d+1)\operatorname{tr}(\Lambda)}{K_{\mathrm{test}}} + \frac{4(1 + 2d + d^2\kappa)\operatorname{tr}(\Lambda)}{K^2} \right]. \tag{10}$$

Therefore, the number of support trajectories K_{test}, the number of training trajectories K, and the horizon T play different roles. Increasing K_{test} reduces the in-context estimation error, while increasing K reduces the training-length bias. Increasing the horizon lengthens each trajectory, but it also increases the number of decision points over which Q-prediction errors can compound. These effects cancel in the second term but not in the first term in (10). We provide the proof of Proposition 1 and additional theoretical results in Appendix E.

5 NUMERICAL EXPERIMENTS FOR PERFORMANCE EVALUATION

We conduct numerical experiments to assess the performance of LLMs in sequential decision-making under MDP, POMDP, and APOMDP settings. We provide additional experiments in Appendix F, including comparisons with benchmarks such as DPT (Lee et al., 2023), robustness tests, alternative few-shot demonstration analysis, and evaluation on the Darkroom task (Lee et al., 2023).

5.1 TASK GENERATION

MDP. We consider an "energy-management" task that models the trade-off between "charging" and "working" using an MDP. The system consists of $\mathrm{E}+1$ discrete energy states $s \in \{0, 1, \ldots, \mathrm{E}\}$, where higher states correspond to greater stored energy. At each decision step, the agent selects one action from $\mathcal{A} = \{0, 1, 2\}$. Actions $a = 0$ and $a = 2$ are two redundant charge actions: both replenish energy, transition from state s to $\min(\mathrm{E}, s+1)$ with probability p, and incur the same small cost c. The work action is $a = 1$. The work action consumes energy, transitioning from state s to $\max(0, s-1)$ with probability p, and yields a reward proportional to the current energy level, $r = s/\mathrm{E}$. The duplicated charge actions enlarge the action space, making action selection harder because two actions correspond to the same physical control. We sample the probability $p \in [0.5, 1)$ uniformly to synthesize training and test tasks.[1] While the theoretical analysis mentioned above studies a linear MDP, our experiments evaluate the proposed framework on tabular MDP settings.

POMDP. The POMDP uses the same state dynamics and reward structure, but the agent receives only noisy observations. Each state emits one of $\mathrm{E}+1$ discrete observation symbols indexed by $\{0, 1, \ldots, \mathrm{E}\}$, where the correct energy state is observed with probability q and the remaining mass is uniformly distributed across other observations. We use $q \in \{0.5, 0.8, 1.0\}$ in our simulation study.

APOMDP. We extend the setting by allowing both partial observability and model ambiguity. The agent faces the same charge-work trade-off but must reason over a set of plausible models. We use a simplified entropy-ball approximation based on Kullback–Leibler (KL) divergence: we first sample a transition-emission model, and additional latent models are Dirichlet perturbations around the first model, accepted only when $\mathrm{KL}(\mathrm{base} \,\|\, \mathrm{candidate}) \le 0.2$. We vary $|\mathcal{M}_\tau| \in \{1, 3, 5\}$.

For each training task τ, we roll out 15 trajectories of length T under its (approximately) optimal policy. We obtain these policies via backward induction for MDPs, belief-state backward induction for POMDPs, and robust backward induction using the α-maximin expected utility (α-MEU) criterion for APOMDPs (Saghafian, 2018).

5.2 PERFORMANCE METRIC

We evaluate performance using the optimality gap:

$$\text{Optimality Gap} = \frac{1}{N_{\text{eval}}} \sum_{\tau=1}^{N_{\text{eval}}} \frac{\mathrm{OPT}_\tau - \mathrm{OPT}_\tau^{\mathrm{Eval}}}{\mathrm{OPT}_\tau}, \tag{11}$$

where N_{eval} denotes the number of test tasks, OPT_τ the reward under the optimal policy, and $\mathrm{OPT}_\tau^{\mathrm{Eval}}$ the reward under the evaluated policy. A smaller gap indicates closer-to-optimal performance.

5.3 ESTIMATION PERFORMANCE

We compute the optimality gap on 100 test tasks by first computing rewards across evaluation trajectories within each task (30 for MDPs/POMDPs, 90 for APOMDPs), then averaging across tasks. Confidence intervals are two-sided Student's t intervals around the mean over test tasks, scaled by its standard error. Evaluation uses two few-shot support trajectories generated by the optimal policy. The ICL-only baseline uses the same pretrained base LLM without supervised fine-tuning.

[1] In all experiments, we set $\mathrm{E} = 9$ and $c = -0.02$.

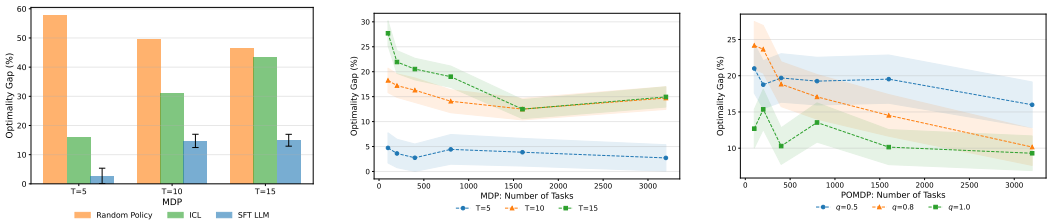

Figure 2: Left: the optimality gaps of the random policy, the ICL policy, and the SFT policy (3,200 training tasks) across planning horizons $T \in \{5, 10, 15\}$ for MDPs. Center: the optimality gaps of the SFT policy versus the number of training tasks for MDPs across planning horizons $T \in \{5, 10, 15\}$. Right: the optimality gaps of the SFT policy versus the number of training tasks for POMDPs with planning horizon $T = 10$ and the values of q equal to 0.5, 0.8, and 1.0. The shaded areas represent the 95% confidence interval.

MDP. Figure 2 (left) shows the optimality gaps of the random policy, the ICL policy, and the SFT policy (3,200 training tasks) across planning horizons $T \in \{5, 10, 15\}$. The random policy achieves optimality gaps of 57.6%, 49.6%, and 46.4% for $T = 5$, 10, and 15, respectively. The ICL policy reduces these gaps to 15.9%, 31.1%, and 43.2%, which provides meaningful gains over random guessing. The SFT policy further reduces the gap to 3%, 15%, and 15% for $T = 5$, 10, and 15, respectively. In particular, we see the largest improvements at longer horizons where in-context learning alone struggles.

Figure 2 (center) shows the optimality gap as a function of the number of training tasks (100, 200, 400, 800, 1,600, and 3,200). For $T = 10$ and 15, the gap decreases as the number of training tasks increases. For example, at $T = 15$, the gap decreases from about 30% to 15%. For $T = 5$, the SFT policy already performs well with few training tasks. The larger improvements for longer horizons indicate that fine-tuning the LLM is valuable for complex, longer-horizon problems.

POMDP. Figure 2 (right) shows the optimality gaps under different observation probabilities with planning horizon $T = 10$. When $q = 1$ (full observability), the gap is smallest. As q decreases, belief states become less informative and the gaps increase. Additional fine-tuning data are especially valuable in partially observed environments, with the fully observed case showing modest gains from additional training tasks.

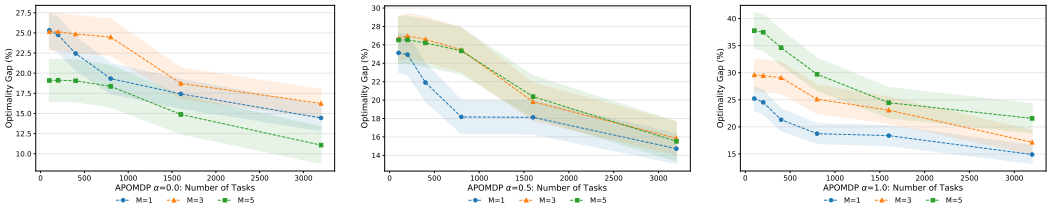

Figure 3: Optimality gaps versus the number of training tasks for APOMDPs. Planning horizon $T = 10$ and $q = 0.8$. The three lines correspond to ambiguity set sizes $|\mathcal{M}_\tau| \in \{1, 3, 5\}$. The three panels correspond to $\alpha = 0.0$ (left), $\alpha = 0.5$ (center), and $\alpha = 1.0$ (right). The shaded areas represent the 95% confidence interval.

APOMDP. Figure 3 shows the optimality gaps for APOMDPs. For a given number of models, the gap decreases as the number of training tasks increases. The effect of ambiguity set size $|\mathcal{M}_\tau|$, however, depends on the ambiguity attitude α. Under the optimistic criterion ($\alpha = 0$), a larger $|\mathcal{M}_\tau|$ expands the maximization space, providing the agent with a richer set of candidate models to exploit. This broader exploration scope facilitates better generalization, and the optimality gap decreases when $|\mathcal{M}_\tau|$ increases. In contrast, under the pessimistic criterion ($\alpha = 1$), a larger $|\mathcal{M}_\tau|$ tightens the worst-case constraint by minimizing over more models and increases the optimality gap. The neutral criterion ($\alpha = 0.5$) balances these effects, and the gap is relatively stable when $|\mathcal{M}_\tau|$ increases. These results suggest that fine-tuning LLMs allows them to handle model ambiguity, and that the benefits depend on the ambiguity attitude and the size of the ambiguity set.

Comparison against DPT. We compare our fine-tuned LLM against DPT (Lee et al., 2023) with 3,200 training tasks. Across all three settings, our approach achieves smaller optimality gaps. In the MDP setting, the gap decreases from 43% (DPT) to 15% (SFT) at $T = 15$. In the POMDP setting, the gap decreases from 25% (DPT) to 10% (SFT) at $q = 0.8$. In the APOMDP setting, our approach maintains a roughly 10-percentage-point advantage across all ambiguity attitudes. In particular, the gaps for DPT grow faster than those of our approach under the pessimistic criterion ($\alpha = 1.0$) when the ambiguity set size increases. We provide more details in Appendix F.1.

Robustness to out-of-distribution (OOD) test conditions. We evaluate the robustness of our approach by testing on parameters outside the training distribution. For MDPs, the model trained with $p \in [0.5, 1)$ and tested on $p \in [0.3, 0.5)$ achieves low out-of-distribution gaps at all three planning horizons ($T = 5$, 10, and 15). For POMDPs, the model trained with $q = 0.8$ achieves nearly identical gaps when tested at $q \in \{0.6, 0.7, 0.9\}$. For APOMDPs, the model trained at $\alpha = 0.5$ transfers well to $\alpha \in \{0.0, 1.0\}$, with gaps increasing slightly. We provide more details in Appendix F.2.

Quality of few-shot support trajectories. Our results suggest that replacing rollouts under the optimal policy with rollouts under the random policy in the few-shot support trajectories affects the performance. The difference is most pronounced at $T = 5$: with optimal support trajectories, the SFT LLM achieves an optimality gap of about 3%, while with random support trajectories the gap remains around 20% even at 3,200 training tasks. We provide more details in Appendix F.3.

Performance on Darkroom tasks. To further evaluate our approach, we use the Darkroom task (Lee et al., 2023), a 10×10 deterministic grid-world with a hidden goal cell. The agent has five actions (the four cardinal moves and a stay action), receives a reward of $+1$ only when it executes the stay on the goal cell, and 0 otherwise. The task is challenging because rewards are sparse and the goal location is unobservable, so the LLM must infer it from the few-shot demonstrations alone. We split the 100 possible goals into 80 fine-tuning goals and 20 held-out test goals. Our approach achieves approximately 95% of oracle performance on 20 test goals, far exceeding the random policy. We provide more details in Appendix F.4.

6 Discussion

We study the ability of LLMs to support sequential decision-making. We propose a supervised fine-tuning framework that adapts a pretrained LLM using parameter-efficient QLoRA and offline, oracle-labeled sequential decision-making data. This process yields a history-conditioned policy that can generalize to new tasks via few-shot support trajectories. Empirically, we move beyond typical fully observed MDPs and evaluate LLMs on two more challenging decision-making environments that allow partial observability and model ambiguity: POMDPs and APOMDPs. Our experiments show that fine-tuned LLMs consistently perform well across all three settings, with the largest gains in long-horizon, partially observed, and model-ambiguous settings. We also compares our approach against DPT across all benchmarks and demonstrate robust generalization to out-of-distribution test conditions. Our theoretical results interpret the fine-tuned attention mechanism as implicitly estimating optimal Q-function parameters, and use this interpretation to create an end-to-end suboptimality bound that separates the in-context estimation error from the training-length bias.

We discuss the limitations of our work in Appendix G. Several promising directions remain for future work. A primary motivation of our work is applying this framework to real-world clinical decision-making using electronic health record data, where offline observational data are abundant. Using RL or other approaches to improve LLM capabilities in understanding causation under ambiguity and proposing effective dynamic treatment recommendations (see, e.g., Saghafian, 2024) can be an important path toward achieving this goal. Furthermore, reducing the reliance on oracle supervision would significantly broaden applicability to domains where optimal policies are hard to obtain. Extending the theoretical and empirical analysis to multimodal settings is another natural next step, as it can enable further progress toward LLM preparation for real-world decision-making applications such as those in healthcare. Finally, exploring whether the natural language capabilities of pretrained LLMs could incorporate human intuition alongside numerical trajectories (e.g., to create powerful centaurs Saghafian 2025; Saghafian & Idan 2024; Dean et al. 2022) may open new avenues for knowledge transfer in complex, high-dimensional environments.

AI USE STATEMENT

Meta's Llama-2-7B is used as the model studied in this work and is fine-tuned with QLoRA for in-context sequential decision-making, as described in Section 3 and Appendix C. This experimental use is part of the proposed method.

Separately, the authors used OpenAI Codex to check and improve the grammar of the manuscript and to assist with LaTeX formatting. Codex and other general-purpose generative AI assistants were not used to generate synthetic data, develop the theoretical or conceptual framework, formulate mathematical claims, provide ingredients for or write proofs, propose or refine hypotheses, design the research methodology or experiments, implement the methods, translate the manuscript, clean or reformat data, perform qualitative or thematic analysis, or interpret the experimental results.

All AI-suggested editorial changes in the paper were manually reviewed and revised by the authors. The authors take full responsibility for the final content of this work.

REFERENCES

Kwangjun Ahn, Xiang Cheng, Hadi Daneshmand, and Suvrit Sra. Transformers learn to implement preconditioned gradient descent for in-context learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-1977.

Yu Bai, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/abs/2306.04637.

Jose Blanchet, Miao Lu, Tong Zhang, and Han Zhong. Double pessimism is provably efficient for distributionally robust offline reinforcement learning: Generic algorithm and robust partial coverage. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901, 2020.

Yuji Cao, Huan Zhao, Yuheng Cheng, Ting Shu, Yue Chen, Guolong Liu, Gaoqi Liang, Junhua Zhao, Jinyue Yan, and Yun Li. Survey on large language model-enhanced reinforcement learning: Concept, taxonomy, and methods. *IEEE Transactions on Neural Networks and Learning Systems*, 36(6):9737–9757, 2025. doi: 10.1109/TNNLS.2024.3497992.

Thomas Carta, Clément Romac, Thomas Wolf, Sylvain Lamprier, Olivier Sigaud, and Pierre-Yves Oudeyer. Grounding large language models in interactive environments with online reinforcement learning. In *International Conference on Machine Learning*, pp. 3676–3713, 2023. URL https://arxiv.org/abs/2302.02662.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *Advances in Neural Information Processing Systems*, 34:15084–15097, 2021.

Sanxing Chen, Xiaoyin Chen, Yukun Huang, Roy Xie, and Bhuwan Dhingra. When greedy wins: Emergent exploitation bias in meta-bandit LLM training. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2509.24923.

Zhenwen Dai, Federico Tomasi, and Sina Ghiassian. In-context exploration-exploitation for reinforcement learning. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2403.06826.

Arlen Dean, Agni Orfanoudaki, Soroush Saghafian, Karen Song, Harini A Chakkera, and Curtiss Cook. Algorithm, human, or the centaur: How to enhance clinical care? *HKS Working Paper No. RWP22-027*, 2022.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. *Advances in Neural Information Processing Systems*, 36:10088–10115, 2023.

Juncheng Dong, Moyang Guo, Ethan X. Fang, Zhuoran Yang, and Vahid Tarokh. In-context reinforcement learning from suboptimal historical data. In *International Conference on Machine Learning*, pp. 14021–14039, 2025. URL https://proceedings.mlr.press/v267/dong25d.html.

Yan Duan, John Schulman, Xi Chen, Peter L Bartlett, Ilya Sutskever, and Pieter Abbeel. RL^2: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*, 2016.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *International Conference on Machine Learning*, pp. 1126–1135, 2017.

Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. *Advances in Neural Information Processing Systems*, 34:20132–20145, 2021.

Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. What can transformers learn in-context? A case study of simple function classes. *Advances in Neural Information Processing Systems*, 35:30583–30598, 2022.

Mohammad Ghavamzadeh, Shie Mannor, Joelle Pineau, and Aviv Tamar. Bayesian reinforcement learning: A survey. *Foundations and Trends in Machine Learning*, 8(5-6):359–483, 2015. doi: 10.1561/2200000049.

Jake Grigsby, Linxi Fan, and Yuke Zhu. AMAGO: Scalable in-context reinforcement learning for adaptive agents. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.09971.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*, 2022.

Garud N. Iyengar. Robust dynamic programming. *Mathematics of Operations Research*, 30(2): 257–280, 2005. doi: 10.1287/moor.1040.0129.

Ying Jin, Zhuoran Yang, and Zhaoran Wang. Is pessimism provably efficient for offline RL? *International Conference on Machine Learning*, pp. 5084–5096, 2021.

Akshay Krishnamurthy, Keegan Harris, Dylan J. Foster, Cyril Zhang, and Aleksandrs Slivkins. Can large language models explore in-context? In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://arxiv.org/abs/2403.15371.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative Q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33:1179–1191, 2020.

Michael Laskin, Luyu Wang, Junhyuk Oh, Emilio Parisotto, Stephen Spencer, Richie Steigerwald, DJ Strouse, Steven Hansen, Angelos Filos, Ethan Brooks, et al. In-context reinforcement learning with algorithm distillation. *arXiv preprint arXiv:2210.14215*, 2022.

Jonathan Lee, Annie Xie, Aldo Pacchiano, Yash Chandak, Chelsea Finn, Ofir Nachum, and Emma Brunskill. Supervised pretraining can learn in-context reinforcement learning. *Advances in Neural Information Processing Systems*, 36:43057–43083, 2023.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Shuang Li, Xavier Puig, Chris Paxton, Yilun Du, Clinton Wang, Linxi Fan, Tao Chen, De-An Huang, Ekin Akyürek, Anima Anandkumar, et al. Pre-trained language models for interactive decision-making. *Advances in Neural Information Processing Systems*, 35:31199–31212, 2022.

Licong Lin, Yu Bai, and Song Mei. Transformers as decision makers: Provable in-context reinforcement learning via supervised pretraining. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=yN4Wv17ss3.

Xiaoyu Liu, Paiheng Xu, Junda Wu, Jiaxin Yuan, Yifan Yang, Yuhang Zhou, Fuxiao Liu, Tianrui Guan, Haoliang Wang, Tong Yu, et al. Large language models and causal inference in collaboration: A comprehensive survey. *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 7668–7684, 2025.

Chenhao Lu, Ruizhe Shi, Yuyao Liu, Kaizhe Hu, Simon Shaolei Du, and Huazhe Xu. Rethinking transformers in solving POMDPs. In *International Conference on Machine Learning*, pp. 33089–33112, 2024. URL https://proceedings.mlr.press/v235/lu24h.html.

Arvind Mahankali, Tatsunori B. Hashimoto, and Tengyu Ma. One step of gradient descent is provably the optimal in-context learner with one layer of linear self-attention. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2307.03576.

Suvir Mirchandani, Fei Xia, Pete Florence, Brian Ichter, Danny Driess, Montserrat Gonzalez Arenas, Kanishka Rao, Dorsa Sadigh, and Andy Zeng. Large language models as general pattern machines. *arXiv preprint arXiv:2307.04721*, 2023.

Amir Moeini, Jiuqi Wang, Jacob Beck, Ethan Blaser, Shimon Whiteson, Rohan Chandra, and Shangtong Zhang. A survey of in-context reinforcement learning. *arXiv preprint arXiv:2502.07978*, 2025. URL https://arxiv.org/abs/2502.07978.

Susan A Murphy. Optimal dynamic treatment regimes. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 65(2):331–355, 2003.

Allen Nie, Yi Su, Bo Chang, Jonathan Lee, Ed H. Chi, Quoc V. Le, and Minmin Chen. EVOLvE: Evaluating and optimizing LLMs for in-context exploration. In *International Conference on Machine Learning*, pp. 46346–46376, 2025. URL https://proceedings.mlr.press/v267/nie25b.html.

Arnab Nilim and Laurent El Ghaoui. Robust control of Markov decision processes with uncertain transition matrices. *Operations Research*, 53(5):780–798, 2005. doi: 10.1287/opre.1050.0216.

Chanwoo Park, Xiangyu Liu, Asuman Ozdaglar, and Kaiqing Zhang. Do LLM agents have regret? A case study in online learning and games. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2403.16843.

Chanwoo Park, Ziyang Chen, Asuman Ozdaglar, and Kaiqing Zhang. Post-training LLMs as better decision-making agents: A regret-minimization approach. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2511.04393.

Sharath Chandra Raparthy, Eric Hambro, Robert Kirk, Mikael Henaff, and Roberta Raileanu. Generalization to new sequential decision making tasks with in-context learning. In *International Conference on Machine Learning*, pp. 42138–42158, 2024. URL https://proceedings.mlr.press/v235/raparthy24a.html.

Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022.

James Robins. A new approach to causal inference in mortality studies with a sustained exposure period—application to control of the healthy worker survivor effect. *Mathematical Modelling*, 7 (9-12):1393–1512, 1986.

Stéphane Ross, Joelle Pineau, Brahim Chaib-draa, and Pierre Kreitmann. A Bayesian approach for learning and planning in partially observable Markov decision processes. *Journal of Machine Learning Research*, 12(48):1729–1770, 2011. URL https://www.jmlr.org/papers/v12/ross11a.html.

Soroush Saghafian. Ambiguous partially observable Markov decision processes: Structural results and applications. *Journal of Economic Theory*, 178:1–35, 2018.

Soroush Saghafian. Ambiguous dynamic treatment regimes: A reinforcement learning approach. *Management Science*, 70(9):5667–5690, 2024.

Soroush Saghafian. *Insight-driven problem solving: Analytics science to improve the world*. Cambridge University Press, 2025.

Soroush Saghafian and Lihi Idan. Effective generative AI: The human-algorithm centaur. *Harvard Data Science Review*, (Special Issue 5), 2024.

Thomas Schmied, Jörg Bornschein, Jordi Grau-Moya, Markus Wulfmeier, and Razvan Pascanu. LLMs are greedy agents: Effects of RL fine-tuning on decision-making abilities. In *International Conference on Learning Representations*, 2026.

Weihao Tan, Wentao Zhang, Shanqi Liu, Longtao Zheng, Xinrun Wang, and Bo An. True knowledge comes from practice: Aligning LLMs with embodied environments via reinforcement learning. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2401.14151.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pp. 35151–35174, 2023. URL https://proceedings.mlr.press/v202/von-oswald23a.html.

Jane X Wang, Zeb Kurth-Nelson, Dhruva Tirumala, Hubert Soyer, Joel Z Leibo, Rémi Munos, Charles Blundell, Dharshan Kumaran, and Matt Botvinick. Learning to reinforcement learn. *arXiv preprint arXiv:1611.05763*, 2016.

Jiuqi Wang, Ethan Blaser, Hadi Daneshmand, and Shangtong Zhang. Transformers can learn temporal difference methods for in-context reinforcement learning. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Pj06mxCXPl.

Mengdi Xu, Yikang Shen, Shun Zhang, Yuchen Lu, Ding Zhao, Joshua Tenenbaum, and Chuang Gan. Prompting decision transformer for few-shot policy generalization. In *International Conference on Machine Learning*, pp. 24631–24645, 2022. URL https://proceedings.mlr.press/v162/xu22g.html.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2023.

Yuexiang Zhai, Hao Bai, Zipeng Lin, Jiayi Pan, Shengbang Tong, Yifei Zhou, Alane Suhr, Saining Xie, Yann LeCun, Yi Ma, and Sergey Levine. Fine-tuning large vision-language models as decision-making agents via reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://arxiv.org/abs/2405.10292.

Jiaqi Zhang, Joel Jennings, Agrin Hilmkil, Nick Pawlowski, Cheng Zhang, and Chao Ma. Towards causal foundation model: On duality between causal inference and attention. *arXiv preprint arXiv:2310.00809*, 2023.

Ruiqi Zhang, Spencer Frei, and Peter L Bartlett. Trained transformers learn linear models in-context. *Journal of Machine Learning Research*, 25(49):1–55, 2024a.

Xiyuan Zhang, Ranak Roy Chowdhury, Rajesh K Gupta, and Jingbo Shang. Large language models for time series: A survey. *arXiv preprint arXiv:2402.01801*, 2024b.

Siyan Zhao, Tung Nguyen, and Aditya Grover. Probing the decision boundaries of in-context learning in large language models. *Advances in Neural Information Processing Systems*, 37:130408–130432, 2024.

Yifei Zhou, Andrea Zanette, Jiayi Pan, Sergey Levine, and Aviral Kumar. ArCHer: Training language model agents via hierarchical multi-turn RL. In *International Conference on Machine Learning*, pp. 62178–62209, 2024. URL https://arxiv.org/abs/2402.19446.

Ilya Zisman, Vladislav Kurenkov, Alexander Nikulin, Viacheslav Sinii, and Sergey Kolesnikov. Emergence of in-context reinforcement learning from noise distillation. In *International Conference on Machine Learning*, pp. 62832–62846, 2024. URL https://proceedings.mlr.press/v235/zisman24a.html.

A Additional related work

In-context prediction and linear attention. Existing research has investigated the ICL capabilities of LLMs in classification and regression tasks and shown that transformer-based architectures can learn general function classes when trained on in-context prompts (Garg et al., 2022; Zhang et al., 2024a; Mirchandani et al., 2023; Zhao et al., 2024). LLMs have also been applied to time series analysis (see, e.g., Zhang et al., 2024b, for a detailed survey). For the theoretical study of ICL, von Oswald et al. (2023) connect linear self-attention to gradient descent on regression problems. Ahn et al. (2023) and Mahankali et al. (2024) characterize how optimal linear-attention predictors implement gradient or preconditioned-gradient updates, while Bai et al. (2023) study a broader class of in-context statistical algorithms and algorithm selection. Our analysis adopts the trained linear-regression predictor of Zhang et al. (2024a) and combines its prediction-error guarantee with a standard performance-difference argument and an on-policy calibration condition. This yields a policy-level decomposition into test-context error and training-prompt-length bias. It complements the algorithm-imitation and temporal-difference analyses of Lin et al. (2024) and Wang et al. (2025). The result concerns a single linear self-attention layer under linear-MDP and Gaussian-feature assumptions; it does not directly analyze the full pretrained LLMs.

LLMs as decision-making agents. Some studies have explored using LLMs directly for decision-making through prompting. ReAct (Yao et al., 2023) generates both reasoning traces and task-specific actions in an interleaved manner, and Huang et al. (2022) use language feedback for embodied planning. Reed et al. (2022) train a single generalist transformer across vision, language, and control tasks. These approaches either use pretrained models directly or train models from scratch. Our study fine-tunes a pretrained LLM for few-shot sequential decision-making from numerical trajectories. Surveys of related studies in LLM-enhanced RL and in-context RL can be found in Cao et al. (2025) and Moeini et al. (2025).

Post-training objectives and exploration. Park et al. (2025) study LLM agents through regret in online learning and games. More recent work improves decision-making through different post-training signals: EVOLvE uses oracle behavior generated by exploration algorithms (Nie et al., 2025), Park et al. (2026) repeatedly select low-regret model rollouts for SFT, Chen et al. (2026) compare expert SFT with RL using several reward designs, and Schmied et al. (2026) fine-tune on self-generated reasoning and environment rewards. These studies examine exploration, exploitation, and the behavioral effects of post-training. Our study shares the oracle-trajectory SFT paradigm with EVOLvE, but distills policies computed for finite-horizon decision models, including history-dependent policies under partial observability and model ambiguity. We apply an autoregressive loss to serialized trajectories and evaluate discounted policy performance given support trajectories. Our emphasis is on how the same offline training approach reproduces policies defined under different information and ambiguity conditions, complementing the exploration and regret questions studied in this literature.

RL fine-tuning of language-model agents. Other studies adapt pretrained language models through interaction with an environment. GLAM grounds LLM policies using online RL (Carta et al., 2023), and TWOSOME aligns LLMs with embodied environments through RL and parameter-efficient adaptation (Tan et al., 2024). ArCHer uses hierarchical multi-turn RL to address delayed rewards and credit assignment (Zhou et al., 2024), while Zhai et al. (2024) fine-tune vision-language models with task rewards. Our study also adapts a pretrained model, but uses a fixed dataset of solver-generated trajectories for SFT. This allows us to study the imitation of policies defined by different decision criteria without optimizing the LLM through online interaction, at the cost of requiring oracle supervision.

Demonstrations and data quality. Prompt-DT uses trajectory prompts built from segments of few-shot demonstrations to condition policies on new tasks (Xu et al., 2022). Raparthy et al. (2024) show how task diversity, stochasticity, and trajectory burstiness affect in-context generalization. Other studies reduce the need for optimal supervision: Zisman et al. (2024) construct learning histories through noise injection, and Dong et al. (2025) use advantage-weighted likelihood training to learn from suboptimal historical trajectories. ICEE performs the exploration–exploitation trade-off at inference time to make in-context policy learning more efficient (Dai et al., 2024). Our setting focuses

on the distillation of optimal or approximately optimal policies and the effect of demonstration quality at evaluation. Our held-out and parameter-shift experiments concern tasks within the evaluated families.

Meta-reinforcement learning and partial observability. Meta-reinforcement learning trains agents that can adapt to new tasks from a shared task distribution. Duan et al. (2016) and Wang et al. (2016) train recurrent networks on sequences of episodes so that the hidden state implicitly encodes a learning algorithm. Finn et al. (2017) instead learns initializations amenable to fast gradient-based adaptation. AMAGO trains long-context transformers with RL for adaptation and memory-dependent tasks (Grigsby et al., 2024). Pretrained language models have also been adapted for interactive decision-making under partial observability (Li et al., 2022). Lu et al. (2024) identify limitations of transformers in solving POMDPs and propose linear recurrent units as an alternative. Our study uses supervised fine-tuning of pretrained language models and varies observation noise within a controlled environment family. Its additional APOMDP dimension concerns uncertainty about the underlying model and the decision maker's attitude toward that uncertainty.

Bayesian and robust models of uncertainty. Bayesian RL represents uncertainty through a prior and posterior over unknown quantities (Ghavamzadeh et al., 2015). Bayes-adaptive POMDPs combine uncertainty over latent states and model parameters with belief tracking and planning (Ross et al., 2011). Robust MDPs instead optimize against sets of plausible transition models (Iyengar, 2005; Nilim & El Ghaoui, 2005), and distributionally robust offline RL accounts for both limited data coverage and model perturbations (Blanchet et al., 2023). We adopt the APOMDP formulation of Saghafian (2018), which combines partial observability with a set of plausible models and an explicit ambiguity attitude through the α-MEU criterion. This formulation allows us to vary observation quality, model ambiguity, and ambiguity attitude separately without specifying a prior over the candidate models. Building on this existing formulation and its numerical solution methods, our contribution is to study the distillation of the resulting policies into an LLM, including how policy gaps vary with ambiguity-set size and attitude.

Offline reinforcement learning. Offline RL approaches learn policies from previously collected data without online interaction (Levine et al., 2020). Some approaches include Conservative Q-Learning (Kumar et al., 2020), which regularizes Q-values to avoid overestimation of out-of-distribution actions, pessimistic offline RL with provable guarantees (Jin et al., 2021), and minimalist approaches that add behavioral cloning regularization (Fujimoto & Gu, 2021). Our study shares the offline data setting with this literature. Rather than learning the LLM policy through explicit value-function optimization, we fine-tune it on oracle trajectories and condition its actions on few-shot supports. The numerical solvers used to obtain these trajectories remain natural methods when the task model is known and tractable. Likewise, DPT (Lee et al., 2023) is a related supervised approach, but differences in architecture and input representation mean that our comparison does not isolate the effect of language pretraining.

Causal inference and LLMs. Causal foundation models have been developed to estimate treatment effects (see, e.g., Zhang et al., 2023), but these studies typically focus on single-time-step settings. A survey of related studies in this stream can be found in Liu et al. (2025). ADTRs connect sequential decision-making under unobservable confounding with ambiguity over plausible causal models (Saghafian, 2024). This connection motivates our interest in partially observed and model-ambiguous decision problems. Our study examines policy distillation in synthetic environments.

B Definitions: MDP and POMDP

We provide the formal definitions for the MDP and POMDP discussed in Section 3.2.

MDP [full observability]. We set $X_t = S_t \in \mathcal{S}_\tau$ and model the task as

$$\mathcal{E}_\tau^{\mathrm{MDP}} = \big\langle \mathcal{S}_\tau,\ \mathcal{A}_\tau,\ P_\tau,\ R_\tau,\ \rho_\tau,\ T_\tau,\ \gamma \big\rangle,$$

where $P_\tau(\cdot \mid s, a)$ is the state-transition kernel, $R_\tau(s, a)$ the one-step reward, and ρ_τ the initial-state distribution. The Markov property implies $\mathbb{P}(S_{t+1} \mid H_t, a_t) = P_\tau(S_{t+1} \mid S_t, a_t)$ and $r_t =$

$R_\tau(S_t, a_t)$. Without loss of optimality, one may restrict to Markov policies $\pi_t(a \mid H_t) = \pi_t(a \mid S_t)$, and to stationary $\pi(a \mid S)$ when the model is time-homogeneous.

POMDP [partial observability]. When we only observe $O_t \in \mathcal{O}_\tau$ generated from a latent state $S_t \in \mathcal{S}_\tau$, we set $X_t = O_t$ and define

$$\mathcal{E}_\tau^{\mathrm{POMDP}} = \big\langle \mathcal{S}_\tau,\ \mathcal{A}_\tau,\ \mathcal{O}_\tau,\ P_\tau,\ Q_\tau,\ R_\tau,\ \rho_\tau,\ T_\tau,\ \gamma \big\rangle,$$

where P_τ is the state-transition kernel and $Q_\tau(o \mid s, a)$ is the observation kernel. It is often convenient to work with the belief vector $b_t \in \Delta(\mathcal{S}_\tau)$ over latent states, which summarizes all information in the history H_t. The belief is updated via a Bayes operator B_τ, i.e., $b_{t+1} = B_\tau(b_t, a_t, o_{t+1})$, which combines the prior belief b_t, the state transition dynamics, and the observation likelihood to produce a new posterior using Bayes' rule. The belief vector b_t serves as a sufficient statistic that restores a Markov property in the belief space. We restrict attention to belief-based policies $\pi_t(a \mid H_t) = \pi_t(a \mid b_t)$. MDPs arise as special cases with full observability by setting $\mathcal{O}_\tau = \mathcal{S}_\tau$ and $Q_\tau(o \mid s, a) = \mathbf{1}\{o = s\}$.

C Supervised fine-tuning details

We fine-tune an open-source LLM using a parameter-efficient QLoRA setup (Dettmers et al., 2023). Specifically, for each linear module $W \in \mathbb{R}^{d_{\mathrm{out}} \times d_{\mathrm{in}}}$ in attention and multi-layer perceptron blocks, we learn a low-rank update $\Delta W = BA$ with $A \in \mathbb{R}^{r \times d_{\mathrm{in}}}$ and $B \in \mathbb{R}^{d_{\mathrm{out}} \times r}$ and keep the original parameters W_0 of the LLM frozen. Here d_{in} and d_{out} are the input and output feature dimensions of the linear module and r is the rank of the low-rank update. The updated parameters are given by $W = W_0 + \Delta W = W_0 + BA$, where $r \ll \min\{d_{\mathrm{in}}, d_{\mathrm{out}}\}$. Only the adapter parameters (A, B) are updated to minimize the loss $\mathcal{L}_{\mathrm{SFT}}(\theta)$ defined in Eq. (1). Because the base model weights remain fixed, we substantially reduce the number of trainable parameters and computational resources required.

Throughout our experiments, we use Meta's Llama-2-7B as the base model, which strikes a favorable balance between capacity and trainability. We use the Paged AdamW 32-bit optimizer with a learning rate of 2×10^{-4}, a cosine learning rate schedule, and a warmup ratio of 0.05. For the low-rank adaptation (LoRA) component, the adapter rank is set to $r = 64$ with LoRA $\alpha = 64$ and dropout of 0.1, applied to all attention and multilayer perceptron projection layers. We train for three epochs, evaluating on the validation set (10% training tasks) at each epoch. All training is conducted on a single NVIDIA A100 GPU with 80 GB of memory using QLoRA with 4-bit NormalFloat (NF4) quantization and bfloat16 mixed-precision computation to reduce memory usage and accelerate convergence. The wall-clock cost of a single fine-tuning run together with its evaluation ranges from a few hours to roughly three to four days, depending on the number of training tasks. APOMDP runs are more expensive than MDP and POMDP runs because of computing the optimal policy by backward induction. Including preliminary and failed runs that did not appear in the final paper, the full project consumed approximately two to three months to run.

Licenses of existing assets. The base model Llama-2-7B is used under the Llama 2 Community License Agreement (Touvron et al., 2023), which permits research use with attribution. We use the QLoRA implementation (Dettmers et al., 2023) and the `bitsandbytes` library, both released under the MIT license, together with HuggingFace `transformers` and `peft` (Apache 2.0) and PyTorch (BSD-3-Clause). The Darkroom task in Appendix F.4 is reimplemented from the description in Lee et al. (2023).

D Detailed pseudocode

We show detailed pseudocode for the key components of our framework: construction of the fine-tuning dataset, the supervised fine-tuning process with QLoRA, and the evaluation procedure.

Algorithm 1 Training and evaluation pipeline

Require: Training tasks.
Ensure: Fine-tuned model parameters θ and performance evaluation.

1: **(Data)** For each $\tau_{\text{train}} \sim \mathcal{T}$, roll out multiple demonstration trajectories $\mathcal{C}_\tau$ under the optimal policy, serialize them into strings s, and collect $\{\mathcal{C}_\tau\}$.
2: **(ICL)** For each test task $\tau_{\text{test}} \sim \mathcal{T}$, construct textual context ctx_τ from few-shot support trajectories and generate actions.
3: **(SFT)** Fine-tune the LLM with QLoRA by minimizing $\mathcal{L}_{\text{SFT}}(\theta)$ using $s \in \{\mathcal{C}_\tau\}$ (Eq. (1)).
4: **(Eval)** For test tasks $\{\tau_{\text{test}}\}$, compute reward and performance gap.

Algorithm 2 Construct training data for supervised fine-tuning

Require: Task distribution $\mathcal{T}$; number of training tasks N_{ft}; number of trajectories K per task; horizon T.
Ensure: Fine-tuning dataset $\{\mathcal{C}_\tau\} = \bigcup_\tau \mathcal{C}_\tau$ of serialized trajectories.

for $i = 1, \ldots, N_{\text{ft}}$ **do**
  Draw a task τ from $\mathcal{T}$ (MDP/POMDP/APOMDP)
  Compute (approximately) optimal policy for τ
  for $k = 1, \ldots, K$ **do**
    Roll out a trajectory of length T: $(X_t, a_t, r_t)_{t=1}^T$
    Serialize the trajectory into a string via a deterministic encoder E
    Add the serialized trajectory to $\mathcal{C}_\tau$
return $\{\mathcal{C}_\tau\}$

Note. A serialized trajectory is a textual encoding of the sequence $(X_t, a_t, r_t)_{t=1}^T$ under a fixed schema defined by the encoder E. For example, E may produce a string of the form `<O_1> 3, <A_1> 1, <R_1> 0.33, <O_2> 2, ...`, where `<O_t>`, `<A_t>`, and `<R_t>` are characters indicating the observation, action, and reward at period t, respectively.

Algorithm 3 Supervised fine-tuning with QLoRA

Require: Base model parameters W_0; adapter rank r; fine-tuning dataset $\{\mathcal{C}_\tau\}$.
Ensure: Fine-tuned model.

1: For $s \in \{\mathcal{C}_\tau\}$, tokenize $s \mapsto (u_1, \ldots, u_L)$
2: For linear module $W \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, parameterize $W = W_0 + BA$ with $A \in \mathbb{R}^{r \times d_{\text{in}}}$ and $B \in \mathbb{R}^{d_{\text{out}} \times r}$, where W_0 is the frozen base model parameter and BA is the trainable low-rank update
3: Take a gradient step on $\theta = (A, B)$ to minimize the loss function:
$$\mathcal{L}_{\text{SFT}}(\theta) = \mathbb{E}_{s \in \{\mathcal{C}_\tau\}} \left[-\sum_{i=1}^{L} \log p_\theta(u_i \mid u_{<i}) \right] \text{ (Eq. (1))}$$
4: **return** Fine-tuned model

Note. Here d_{in} and d_{out} are the input and output feature dimensions of the linear module.

Algorithm 4 Evaluation via rollouts

Require: LLM model; task distribution $\mathcal{T}$; horizon T; number of trajectories K_{eval} per test task.
Ensure: Reward and performance gap.

1: Draw a task τ_{test} from $\mathcal{T}$ and compute (approximately) optimal policy for τ_{test}
2: Roll out few-shot support trajectories $\mathcal{D}_\tau$
3: Serialize trajectories into a textual context $\text{ctx}_\tau = \mathsf{E}(\mathcal{D}_\tau)$
4: **for** $i = 1, \ldots, K_{\text{eval}}$ and $t = 1, \ldots, T$ **do**
  Form the input (H_t, ctx_τ) and decode the next action token(s) to obtain a_t
  Execute a_t in τ_{test}; observe reward r_t and next observation
  Update history $H_{t+1} \leftarrow H_t \cup \{(a_t, r_t, X_{t+1})\}$
8: **return** Reward and performance gap

Note. We restrict the outputs of the LLM to action tokens and select the one with the highest probability.

E ADDITIONAL THEORETICAL RESULTS

E.1 PROOF OF PROPOSITION 1

Proof. For each time t and state s, greediness of $\widehat{\pi}_t$ with respect to $\widehat{Q}_t$ implies

$$V_t^*(s) - Q_t^*(s, \widehat{\pi}_t(s)) = Q_t^*(s, \pi_t^*(s)) - Q_t^*(s, \widehat{\pi}_t(s))$$
$$\leq \left|\widehat{Q}_t(s, \pi_t^*(s)) - Q_t^*(s, \pi_t^*(s))\right| + \left|\widehat{Q}_t(s, \widehat{\pi}_t(s)) - Q_t^*(s, \widehat{\pi}_t(s))\right|.$$

The finite-horizon performance-difference identity gives

$$J_{\tau_{\text{new}}}(\pi^*) - J_{\tau_{\text{new}}}(\widehat{\pi}) = \sum_{t=1}^{T} \gamma^{t-1} \mathbb{E}_{s_t \sim d_t^{\widehat{\pi}}} \left[V_t^*(s_t) - Q_t^*(s_t, \widehat{\pi}_t(s_t))\right].$$

Combining the previous two displays, applying Jensen's inequality, and using (7) for both π_t^* and $\widehat{\pi}_t$ yields

$$\mathbb{E}\left[V_t^*(s_t) - Q_t^*(s_t, \widehat{\pi}_t(s_t))\right] \leq 2\sqrt{C_{\text{on}} \varepsilon_Q(M, N)}.$$

Summing the geometric series over $t = 1, \ldots, T$ gives (8). Equation (9) follows by substituting (6). □

E.2 NUMERICAL VALIDATION OF PROPOSITION 1

To complement the analytical bounds, we run a simulation that tests the simulation-lemma step of Proposition 1. The objective is to verify that, even when ε_Q is measured empirically rather than upper-bounded analytically, the cumulative gap satisfies

$$J_{\tau_{\text{new}}}(\pi^*) - J_{\tau_{\text{new}}}(\hat{\pi}) \;\leq\; C_{T,\gamma} \sqrt{C_{\text{on}}} \sqrt{\widehat{\varepsilon}_Q}, \tag{12}$$

where $\widehat{\varepsilon}_Q$ is the empirical mean of $(\hat{Q} - Q^*)^2$ along the same trajectory used to evaluate the gap.

Setup. We consider a linear MDP with feature dimension $d = 10$, $|\mathcal{A}| = 5$ actions per step, horizon $T = 10$, and discount $\gamma = 0.95$. For each task τ we draw $w_\tau^* \sim \mathcal{N}(\mathbf{0}_d, I_d)$ independently. Per-step candidate features are i.i.d. $\phi_{t,a} \sim \mathcal{N}(\mathbf{0}_d, \Lambda)$, and the in-context demonstrations are noiseless pairs $(x_i, \langle w_\tau^*, x_i \rangle)$ with $x_i \sim \mathcal{N}(\mathbf{0}_d, \Lambda)$. The closed-form predictor $\Gamma^{-1}\big(\frac{1}{M}\sum_i y_i x_i\big)$ from Eq. (5) is applied with $\Gamma = (1+1/N)\Lambda + (\operatorname{tr}\Lambda/N) I_d$. The empirical gap is $\widehat{\text{gap}} = \sum_{t=1}^{T} \gamma^t [Q^*(s_t, \pi^*(s_t)) - Q^*(s_t, \hat{\pi}(s_t))]$, and $\widehat{\varepsilon}_Q$ is the mean of $(\hat{Q} - Q^*)^2$ over the $T \cdot |\mathcal{A}| = 50$ query points. All quantities are averaged over 500 independent tasks per configuration. We jointly vary three key parameters that could affect the gap and ε_Q:

- Support trajectory length $M \in \{10, 20, 50, 100, 200, 500, 1000\}$, which controls $\widehat{\varepsilon}_Q$ along each curve.
- Training length $N \in \{100, 1000, 10000\}$, which enters through Γ^{-1} and controls the training-length bias.
- Feature covariance via a diagonal Λ with log-spaced eigenvalues between $1/\kappa$ and 1, for $\kappa \in \{1, 5, 25\}$ (isotropic, mild, and strong anisotropy).

Results. Figure 4 plots the empirical gap against $\widehat{\varepsilon}_Q$ over all $7 \times 3 \times 3 = 63$ configurations, with one panel per Λ and one curve per N. We see that every empirical curve lies strictly below the dashed reference line (12): the simulation-lemma step holds in every configuration, including the strongly anisotropic regime $\kappa = 25$ in which $\operatorname{tr}(\Lambda^{-1})$ is largest.

E.3 JOINT SAMPLE COMPLEXITY FOR AN ε-OPTIMAL POLICY

The decomposition (10) gives a direct sufficient condition for achieving a target policy quality.

Proposition 2 (Sample complexity for ε-optimal policies)**.** *Fix $\varepsilon > 0$. Under the conditions of Proposition 1, with $\gamma \to 1$, the induced policy $\widehat{\pi}$ satisfies*

$$\mathbb{E}\left[J_{\tau_{\text{new}}}(\pi^*) - J_{\tau_{\text{new}}}(\widehat{\pi})\right] \leq \varepsilon$$

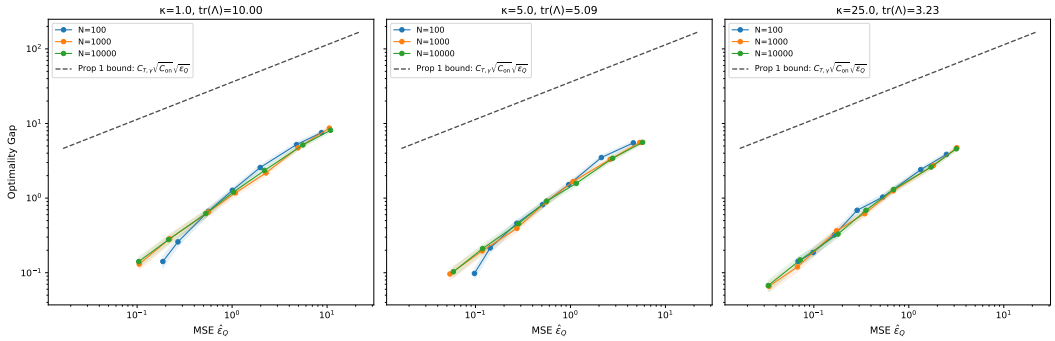

Figure 4: Empirical gap versus measured $\widehat{\varepsilon}_Q$ across Λ, training length N, and support trajectory length M. All curves remain strictly below the dashed reference line $C_{T,\gamma}\sqrt{C_{\mathrm{on}}}\sqrt{\widehat{\varepsilon}_Q}$. The shaded areas represent the 95% confidence intervals.

whenever

$$K_{\mathrm{test}} \geq \frac{8C_{\mathrm{on}}T(d+1)\,\mathrm{tr}(\Lambda)}{\varepsilon^2} \qquad \textit{and} \qquad K \geq \frac{\sqrt{8C_{\mathrm{on}}(1+2d+d^2\kappa)\,\mathrm{tr}(\Lambda)}}{\varepsilon}. \tag{13}$$

In particular, for isotropic features $\Lambda = cI_d$ *with* $c = \Theta(1)$,

$$K_{\mathrm{test}} = O\left(\frac{C_{\mathrm{on}}Td^2}{\varepsilon^2}\right), \qquad K = O\left(\frac{\sqrt{C_{\mathrm{on}}}\,d^{3/2}}{\varepsilon}\right). \tag{14}$$

Proof. Set each of the two terms in (10) to be at most $\varepsilon^2/2$ and solve for K_{test} and K. The isotropic specialization follows by substituting $\Lambda = cI_d$, which gives $\kappa = 1$ and $\mathrm{tr}(\Lambda) = cd$. □

Proposition 2 highlights two qualitative differences between support trajectories and training trajectories. First, the required number of support trajectories grows with the rollout horizon T, while the required number of training rollouts per task does not after substituting $N = KT$. Second, K_{test} scales as $1/\varepsilon^2$, whereas K scales as $1/\varepsilon$. This result suggests that, in the regime captured by the theory, increasing the amount of training data per task could be more sample-efficient than relying only on additional in-context demonstrations.

E.4 Qualitative implications for POMDPs and APOMDPs

The linear MDP assumptions above are restrictive, but the decomposition in Proposition 1 still gives useful qualitative guidance for the partially observed and ambiguous settings studied in our numerical experiments.

Partial observability. As discussed earlier, in POMDPs a sufficient statistic for decision-making is the belief state b_t. If we replace $\phi(s,a)$ with belief-action features $\phi(b,a)$, the same analysis applies formally only when the optimal belief-state Q-function is linear in those features and when the induced feature covariance is stable across support trajectories. As observation fidelity decreases, belief updates become noisier and the effective feature covariance can become more ill-conditioned, increasing κ and the on-policy mismatch constant C_{on}. Proposition 1 then predicts larger optimality gaps and a larger benefit from additional training data.

Model ambiguity. In APOMDPs, the α-MEU optimal Q-function need not be linear in a fixed feature map, and thus Proposition 1 does not apply directly. Nevertheless, the same error decomposition suggests two relevant mechanisms. First, larger ambiguity sets can increase the complexity of the effective Q-function. Second, pessimistic or optimistic ambiguity criteria can change the induced visitation distribution, thereby changing the effective covariance and the on-policy mismatch constant. These effects are consistent with the finding of our numerical experiments that ambiguity size and ambiguity attitude jointly affect the optimality gap.

F ADDITIONAL EXPERIMENTAL RESULTS

F.1 COMPARISON AGAINST DPT

We construct the DPT training dataset using the trajectories $\{\mathcal{C}_\tau\}$ generated by Algorithm 2. Instead of using the serialized trajectories directly, we follow the approach in Lee et al. (2023) to extract sequences of observations, actions, next observations, and rewards from the trajectories and organize them into a tabular format. Specifically, we construct the context based on H_t and use the optimal action at time t as the supervised label. We follow the same training procedure as described in Lee et al. (2023) to train DPT. During evaluation, we use the same test tasks to roll out trajectories and compute the optimality gap. Figures 5 and 6 show the optimality gap across all three settings.

MDP. In the MDP setting (Figure 5, left), our approach achieves smaller optimality gaps than DPT across all planning horizons. The advantage is pronounced for longer horizons: at $T = 10$, our approach reduces the gap from 26% to 15%, and at $T = 15$ from 43% to 15%. At $T = 5$, both approaches perform well, though our approach yields a smaller gap (3% versus 7%).

POMDP. In the POMDP setting (Figure 5, right), our approach outperforms DPT across all observation regimes. The gap is pronounced at $q = 0.8$ and $q = 1.0$, where our approach achieves optimality gaps of roughly 10% and 9% versus 25% and 26% for DPT, respectively. Even under the most challenging condition ($q = 0.5$), our approach has a smaller gap (16% versus 22%).

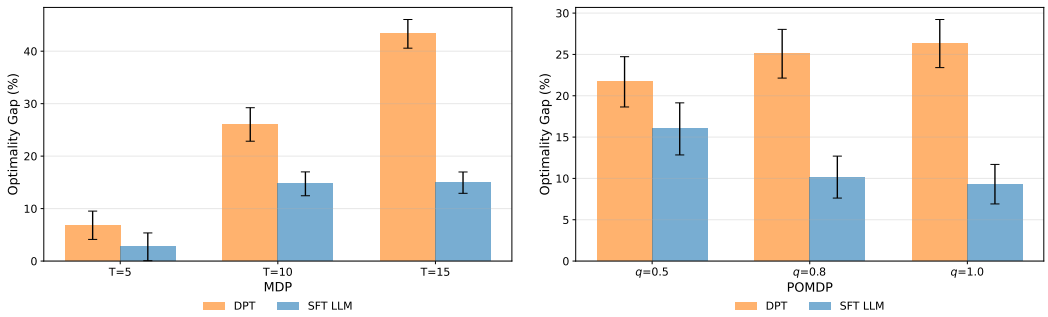

Figure 5: Optimality gaps of our approach versus DPT with 3,200 training tasks for MDP (left) and POMDP (right). In the MDP panel, the three groups correspond to planning horizons $T \in \{5, 10, 15\}$. In the POMDP panel, $T = 10$ and the three groups correspond to $q \in \{0.5, 0.8, 1.0\}$. Error bars represent 95% confidence intervals.

APOMDP. In the APOMDP setting (Figure 6), our approach achieves smaller optimality gaps across all combinations of α and $|\mathcal{M}_\tau|$. Under $\alpha = 0.5$, our approach maintains a consistent 10-percentage-point advantage. Under $\alpha = 1.0$, the gaps for DPT grow faster (from 26% to 37%) than those for our approach (15% to 22%) when $|\mathcal{M}_\tau|$ increases.

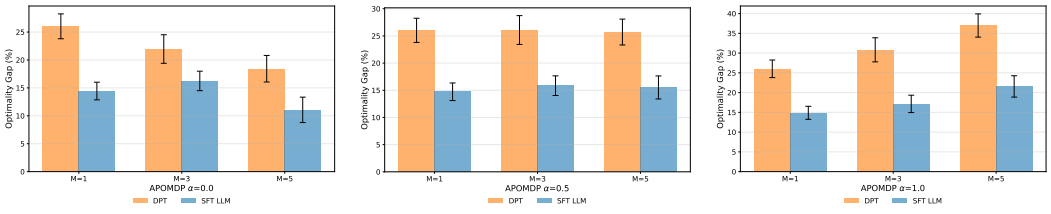

Figure 6: Optimality gaps of our approach versus DPT with 3,200 training tasks for APOMDP. In all three panels, $T = 10$ and $q = 0.8$. The three panels correspond to $\alpha = 0.0$ (left), $\alpha = 0.5$ (center), and $\alpha = 1.0$ (right). Within each panel, the three groups correspond to $|\mathcal{M}_\tau| \in \{1, 3, 5\}$. Error bars represent 95% confidence intervals.

F.2 Robustness to OOD test conditions

We examine generalization beyond the training distribution with the model fixed after fine-tuning on 3,200 tasks. Figure 7 shows optimality gaps for both in-distribution and OOD conditions.

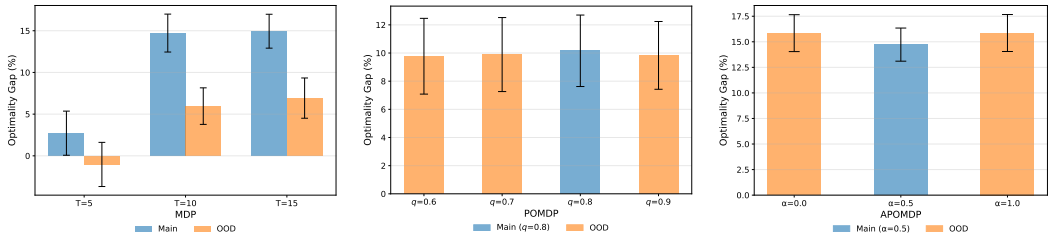

Figure 7: Robustness of the fine-tuned LLM to the OOD test conditions with 3,200 training tasks for MDP (left), POMDP (center), and APOMDP (right). Blue bars: in-distribution. Orange bars: OOD. Error bars represent 95% confidence intervals.

MDP. The model trained on $p \in [0.5, 1)$ is tested on $p \in [0.3, 0.5)$. The OOD gaps are small across all conditions: zero at $T = 5$, 6% at $T = 10$, 7% at $T = 15$. The small OOD gaps may reflect that these test tasks are easier than the training tasks because smaller transition probabilities make state changes less frequent.

POMDP. The model trained on $q = 0.8$ is tested on $q \in \{0.6, 0.7, 0.9\}$. The OOD gaps are around 10% in all conditions, suggesting the robustness to shifts in observation probability.

APOMDP. The model trained on $\alpha = 0.5$ is tested on $\alpha \in \{0.0, 1.0\}$. The OOD gaps are only marginally above the in-distribution gap.

F.3 Quality of few-shot support trajectories

We examine whether demonstration quality affects performance by replacing rollouts under the optimal policy with rollouts under the random policy in the few-shot support trajectories. Figure 8 shows the optimality gaps for both conditions across all three planning horizons.

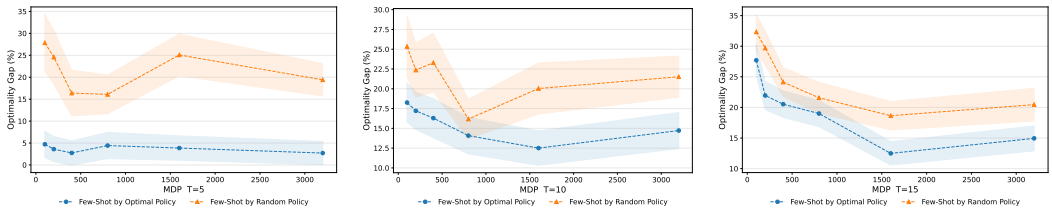

Figure 8: Optimality gaps for few-shot support trajectories generated by the optimal policy (blue) and by the random policy (orange) under an MDP setting with $T = 5$ (left), $T = 10$ (center), and $T = 15$ (right). The x-axis is the number of training tasks. The shaded areas represent the 95% confidence interval.

The effect is pronounced at $T = 5$. The gap is small (roughly 3%–5%) when the few-shot support trajectories are generated by the optimal policy, while it is around 20% when they are generated by the random policy. At $T = 10$ and $T = 15$, we also see a gap of around 15% for optimal support trajectories and 20% for random support trajectories. These results suggest that demonstration quality could affect the performance of the fine-tuned LLM, and that the model can leverage high-quality demonstrations to achieve better performance. The gaps under a large number of training tasks suggest that the model may still benefit from high-quality support trajectories, even though we could provide more training tasks.

F.4 Performance on Darkroom tasks

We apply our approach to the Darkroom task (Lee et al., 2023), a 10×10 grid-world with hidden goals, sparse rewards (reward $+1$ at the goal and 0 otherwise), and horizon $T = 100$. We use 80 goal locations for fine-tuning and 20 for evaluation. Training trajectories are a mixture of 10 oracle and 100 random rollouts per goal. Test evaluations use five rollouts per goal, starting from $(0, 0)$. The few-shot support trajectories are constructed from oracle rollouts for each test goal. The outcome of interest is the cumulative reward over the horizon. Figure 9 shows that the random policy achieves near-zero reward, while the oracle achieves a cumulative reward of approximately 92, and our fine-tuned LLM achieves approximately 87 (95% of the oracle), suggesting the effectiveness of our approach in a challenging, partially observed environment with sparse rewards.

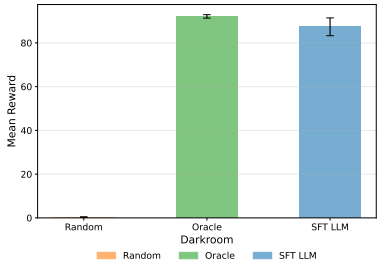

Figure 9: Cumulative reward of the random policy, oracle, and our fine-tuned LLM for Darkroom tasks. Error bars represent 95% confidence intervals.

G Limitations

Our work has several limitations. First, supervised fine-tuning incurs nontrivial computational cost, which constrains the LLM sizes and the breadth of hyperparameters we are able to explore. Second, our framework relies on access to an optimal (or near-optimal) action oracle, which may be hard to compute once the problem becomes complex. Third, our theoretical analysis focuses on linear MDPs with a single-layer linear self-attention model; extensions to deeper architectures and to partially observed or model-ambiguous settings remain open (see Appendix E for partial qualitative results). Finally, our empirical evaluation is conducted on synthetic MDP, POMDP, and APOMDP settings, and validating the approach on real-world clinical datasets is an important next step.