HeFLO: Flow-Regularized Offline Policy Learning under System Heterogeneity and Its Convergence Analysis

Anonymous authors
Paper under double-blind review

Abstract

Offline reinforcement learning (RL) learns control policies from pre-collected data without environment interaction, which is essential where online exploration is costly or unsafe. In practice such data are gathered from heterogeneous systems that share a dynamical structure but may differ in physical parameters. Additionally, the experts may themselves be specialized to particular configurations. We propose HeFLO: Heterogeneous FLow-regularised Offline policy learning, which minimises an infinite-horizon average cost over a bounded range of parametric uncertainty with flow-based behaviour-cloning regularisation. This is a domain-randomisation objective, and HeFLO is its offline, model-free, demonstration-regularised counterpart. For heterogeneous linear quadratic regulators we prove that a regulariser above an explicit and initialisation-dependent threshold confines every iterate to the stability radius of the demonstrated behaviour, so the closed loop is stable for every configuration throughout training. Additionally, the iterates converge geometrically to a neighbourhood whose radius depends on the demonstration set. Empirically, HeFLO reaches near-expert performance on average over the configuration family across heterogeneous MuJoCo environments.

1 Introduction

Actor-critic (AC) methods (Konda & Tsitsiklis, 1999; Schulman et al., 2015) combine an actor that represents the policy with a critic that evaluates it. When paired with function approximation (Haarnoja et al., 2018; Fujimoto et al., 2018), these methods are suitable for continuous state-action Markov decision processes (MDPs) of automatic control (Kiumarsi et al., 2017; Recht, 2019). AC methods are in general online, which means they require continuous interaction with the environment to update the policy and value estimates. However, such online interaction can be sample-inefficient, and in safety-critical settings such as autonomous systems (Kiran et al., 2021) and healthcare (Coronato et al., 2020) it can be infeasible, since exploration may cause irreversible failures (Berkenkamp et al., 2017; Garcıa & Fernández, 2015). On the other hand, offline RL (Lange et al., 2012; Levine et al., 2020) and its variations, such as flow Q-learning (Park et al., 2025), learn primarily from pre-collected data, usually expert demonstrations. These demonstrations are often easier to obtain and provide rich information about system behaviour, especially when the dynamics are unknown or only partially characterised.

1.1 Technical challenges and contributions

Two gaps in the current research remain relatively unexplored. First, offline algorithms tend to perform poorly on heterogeneous systems, where small variations in parameters such as masses, damping coefficients, actuator gains, or friction may not alter the dynamical structure but can shift the deployment environment away from the training distribution. Training a single policy against a distribution of such parameters is known as domain randomization (DR) (Tobin et al., 2017). The theoretical foundations of DR have recently been studied by Fujinami et al. (2025) for control and by Chen et al. (2022) for sim-to-real transfer. However, a gap in this area, which we aim to address, is the offline case, where the randomization is fixed in the data and cannot be re-drawn.

We propose HeFLO: Heterogeneous FLow-regularised Offline policy learning and analyse it for linear quadratic regulators (LQR). Our contributions:

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.

1. **Infinite-horizon average-cost formulation.** We minimise an average expected cost with flow-based behaviour-cloning regularisation. Unlike Park et al. (2025), this yields a tractable policy gradient (Silver et al., 2014) and avoids differentiating the Q-function, which is numerically delicate (D'Oro & Jaśkowski, 2020). The average-cost criterion reflects steady-state behaviour rather than a transient weighted by a discount factor.
2. **An explicit account of the offline gap.** We maintain a clear distinction between the control deployment cost and the offline surrogate. In Lemma 2, we show that for heterogeneous LQR the two induce the same policy-gradient matrix, weighted by different state covariances. This lets us isolate the offline error into interpretable terms, which we carry through the analysis.
3. **Convergence and stability.** We give what is, to our knowledge, the first analysis of convergence and closed-loop stability for policy-gradient offline RL with behavioural regularisation under DR. Unlike previous studies on DR, which are model-based and can sample the randomisation distribution freely (Fujinami et al., 2025; Chen et al., 2022), our analysis addresses the offline case, in which the randomisation is fixed in the data.
4. **Stability from the regulariser, and its price.** We derive an explicit threshold for α, which is the weight of the behavioural-cloning regulariser. Above that threshold, the policy iterates remain within a stable region around the demonstrated behaviour, ensuring closed-loop stability (Theorem 1(i)).

1.2 Related Work

Offline RL and learning from demonstration. The main challenge in offline RL is the overestimation of Q-function due to distributional shifts (Fujimoto et al., 2019). This issue can be addressed by either penalising actions that are out of distribution (Conservative Q-Learning, CQL; Kumar et al., 2020) or by avoiding them (Implicit Q-Learning, IQL; Kostrikov et al., 2021). An alternative approach involves constraining the policy to align with the behaviour distribution. This can be achieved through behaviour cloning (Torabi et al., 2018) combined with methods such as DDPG (Nair et al., 2018), natural policy gradients (Rajeswaran et al., 2017), or TD3 (Fujimoto & Gu, 2021). These methods clone the exact actions of demonstrators, which limits the expressiveness of the model as datasets grow (Collaboration et al., 2023; Mandlekar et al., 2021). On the other hand, flow-based models, such as FQL (Park et al., 2025), provide a different approach by using a flow model as a regulariser. Additionally, diffusion models can serve a similar purpose (Wang et al., 2023; Hansen-Estruch et al., 2023; Ajay et al., 2023). Note that our stability analysis does not depend on which generative family provides the anchor (see Section 4.5(vi)).

Robustness, heterogeneity and domain randomization. Robust MDPs optimize worst-case performance over an uncertainty set, as discussed in previous works (Iyengar, 2005; Nilim & El Ghaoui, 2005), including offline settings (Panaganti et al., 2022; Wang et al., 2024). In contrast, domain randomization (Tobin et al., 2017) involves randomizing parameters to minimize the average performance, which aligns with our objective outlined in equation 3.2. Our data generation process (Algorithm 2) applies domain randomization to the demonstrators. Furthermore, Fujinami et al. (2025) examine a linear quadratic form and demonstrate that it achieves the optimal $1/N$ decay of excess cost. This result provides a strong argument for preferring the average-case objective rather than pursuing a worst-case formulation in our study. However, that study is model-based, and its policy-gradient procedure masks the gradients of unstabilized scenarios, which leaves it without a convergence guarantee. In a different study, Chen et al. (2022) model the simulator as an MDP with unobserved parameters, offering bounds on the sim-to-real gap and emphasising the importance of history-dependent policies.

A more detailed discussion on the related work can be found in Appendix B.

2 Background

Notation. $\|G\|$, $\|G\|_F$, $\rho(G)$, $\mathrm{Tr}(G)$ are the operator norm, Frobenius norm, spectral radius and trace. $G \succ (\succeq) 0$ denotes positive (semi-)definiteness. Throughout, $c(x,u)$ is a *cost* and every objective is *minimized*, whereas Gymnasium reports *rewards*; we therefore set $c = -r$ when converting. Tables and learning curves are in the reward convention (higher is better), all analysis in the cost convention. An MDP is $(\mathcal{X}, \mathcal{U}, P, c, \nu_0)$ with initial-state distribution ν_0, with $\mathcal{X} \subseteq \mathbb{R}^{n_x}, \mathcal{U} \subseteq \mathbb{R}^{n_u}$. Furthermore, actor–critic methods, offline RL and flow matching are reviewed in Appendix A.

For a parameterized stochastic policy π_ζ with parameter ζ, the infinite-horizon time-average cost is given by

$$J(\zeta) = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}_{x_0\sim\nu_0,\, u_t\sim\pi_\zeta(\cdot|x_t)}[c(x_t, u_t)]. \tag{2.1}$$

Furthermore, the associated action-value function, $Q_{\pi_\zeta}(x, u) = \sum_{t\geq 0} \mathbb{E}[c(x_t, u_t) - J(\zeta) \mid x_0 = x, u_0 = u]$, is defined up to an additive constant and convergent because each summand is a deviation from the average cost. Assuming stationarity (Puterman, 2014) gives a steady-state state distribution d^ζ, and the actor parameter is updated by policy gradient (Sutton et al., 1999; Silver et al., 2014) as follows

$$\nabla_\zeta J(\zeta) = \mathbb{E}_{x\sim d^\zeta,\, u\sim\pi_\zeta(\cdot|x)}\big[Q_{\pi_\zeta}(x, u)\nabla_\zeta \log \pi_\zeta(u|x)\big], \quad \zeta_{k+1} = \zeta_k - \gamma_k \nabla_\zeta J(\zeta_k). \tag{2.2}$$

Offline RL (Levine et al., 2020; Prudencio et al., 2023) learns ζ from a fixed dataset $\mathcal{D}$ of transitions without further interaction with the environment. For flow-based behaviour cloning (Lipman et al., 2024; Liu et al., 2022) a vector field v_{θ_b} is trained by

$$\mathcal{L}_{\text{Flow}}(\theta_b) = \mathbb{E}_{x,a^1(=u)\sim\mathcal{D},\; a^0(=z)\sim\mathcal{N}(0,I),\; t\sim\text{Unif}[0,1]}\big[\|v_{\theta_b}(t, x, a^t) - (a^1 - a^0)\|_2^2\big], \tag{2.3}$$

where $a^1 = u$ is the recorded action and $a^t = (1-t)a^0 + ta^1$ interpolates between Gaussian noise z and that action. Integrating the vector field from $t = 0$ to $t = 1$, with initial condition $a^0 = z$, defines the action map $\mu_{\theta_b}(x, z)$ from noise z to an action u.

3 Offline Policy Learning with Flow Regularisation for Heterogeneous MDPs

In this section, we introduce our approach for offline policy learning in heterogeneous MDPs. We first formalise the heterogeneous MDP setting and then describe how flow-based regularisation can be incorporated into offline policy learning to handle the variability across different system configurations.

3.1 Heterogeneous MDP

The dynamics of each system are

$$x_{t+1} = f(x_t, u_t, \delta, \omega_t), \tag{3.1}$$

where $\delta \sim \rho_\delta$ captures heterogeneity and ω_t is i.i.d. process noise. We assume bounded heterogeneity, i.e., $\delta \in \mathcal{S}_\delta$ almost surely for a compact $\mathcal{S}_\delta$.

Scope, and the status of δ. (i) For numerical simulations, δ is drawn once per system and held fixed across all its trajectories, rather than resampled at each transition, thereby preserving the time-invariance of the closed-loop dynamics. (ii) δ is assumed to be unknown when learning offline. δ affects the policy only implicitly, through the observed states. (iii) Heterogeneity is *parametric, not structural*: all configurations share the dynamics function f and the same state and action spaces, and differ only through δ. Structural differences, such as changes in actuation topology, observation space, or contact mode, are outside the scope of this study.

Dataset construction by domain randomization. We sample m distinct configurations $\delta_1, \ldots, \delta_m$ from ρ_δ and train a specialized expert for each configuration (Algorithm 2) for numerical simulations. This yields the dataset $\mathcal{D} = \{d_j\}_{j=1}^m$, where $d_j = \big\{(x_t^{(i)}, u_t^{(i)}, c_t^{(i)}, x_{t+1}^{(i)})\big\}_{i=1}^{N_d}$ contains the N_d transitions collected from the expert for configuration δ_j, so that $N_\mathcal{D} = mN_d$ in total. This procedure implements domain randomization during data collection, with randomization distribution ρ_δ. Unlike the usual setting under DR (Fujinami et al., 2025; Chen et al., 2022), randomization is confined to the data-collection phase: the learner has access only to the resulting fixed dataset, which does not record the values of δ_j, and cannot sample additional configurations.

Deployment cost and what an offline learner can evaluate. Our objective is the expected long-run average cost at deployment,

$$J(\zeta) = \mathbb{E}_\delta\Big[\mathbb{E}_{x\sim d_\delta^\zeta,\, u\sim\pi_\zeta(\cdot|x)}[c(x, u)]\Big], \tag{3.2}$$

where d_δ^ζ denotes the steady-state state distribution generated by π_ζ for configuration δ. Once $x \sim d_\delta^\zeta$, the one-step cost is time invariant, so under the stationarity assumption, equation 3.2 already equals the time average equation 2.1 per configuration. This objective cannot be evaluated directly by rolling out the current policy, since further environment interaction is unavailable in the offline setting. Instead, the offline dataset $\mathcal{D}$ provides an empirical state distribution $d_\mathcal{D}$. Note that one can follow the standard offline actor–critic practice (Fujimoto et al., 2019; Kumar et al., 2020) and use the following surrogate

$$J_\mathcal{D}(\zeta) = \mathbb{E}_{\delta,\, x\sim d_\mathcal{D},\, u\sim\pi_\zeta(\cdot|x)}\big[Q_{\pi_\zeta}(x,u)\big]. \tag{3.3}$$

In general, $J_\mathcal{D} \neq J$: the two differ exactly in the state distribution over which the critic is averaged. This distribution mismatch is a central challenge in offline RL, and it motivates a behavioural regulariser that keeps π_ζ close to the behaviour policy, thereby limiting the shift between d_δ^ζ and $d_\mathcal{D}$. Note that our Theorem 1 bounds the deployment cost J, rather than the surrogate $J_\mathcal{D}$.

3.2 Objective and update direction

Following Park et al. (2025) at a high level, we first train v_{θ_b} by equation 2.3. Since solving the generating ODE at every step is computationally expensive, the flow policy is used as a regulariser rather than being deployed. With $\mathcal{R}(\zeta) := \mathbb{E}_{\delta,\, x\sim d_\mathcal{D},\, \eta\sim\mathcal{N}(0,\Sigma_\eta)}\|\pi_\zeta(x,\eta) - \mu_{\theta_b}(x,\eta)\|_2^2$, the objective is to minimize the following loss

$$\mathcal{L}(\zeta) = J(\zeta) + \tfrac{\alpha}{2}\mathcal{R}(\zeta), \qquad \alpha > 0. \tag{3.4}$$

Here $d_\mathcal{D}$ is the *empirical* state measure of the realized $\mathcal{D}$. The noise η plays the same role as the standard Gaussian noise (z) in the flow model (2.3). We would like to mention two points about η here. First, the same noise draw is fed to both maps, so $\mathcal{R}$ compares the two policies pathwise, which makes it an exact quadratic in ζ for linear one-step actors. Second, μ_{θ_b} expects standard noise whereas the actor's noise has covariance Σ_η, which is required for the correct scaling of the policy's exploration. We therefore pass $\Sigma_\eta^{-1/2}\eta$ to the flow model and choose Σ_η to match the one-step policy covariance. To keep the notation light, $\mu_{\theta_b}(x,\eta)$ abbreviates $\mu_{\theta_b}(x,\Sigma_\eta^{-1/2}\eta)$ from here on, in the main text and in the appendices alike.

Since J cannot be evaluated directly from $\mathcal{D}$, HeFLO constructs an update direction from the offline dataset rather than using the exact gradient of $\mathcal{L}$. Its first component is the surrogate policy-gradient term

$$g(\zeta) = \mathbb{E}_{\delta,\, x\sim d_\mathcal{D},\, u\sim\pi_\zeta(\cdot|x)}\big[Q_{\pi_\zeta}(x,u)\nabla_\zeta \log \pi_\zeta(u|x)\big], \tag{3.5}$$

which is obtained by replacing d_δ^ζ in equation 2.2 with $d_\mathcal{D}$, as in related offline AC methods (Fujimoto & Gu, 2021; Kumar et al., 2020; Park et al., 2025). This substitution generally introduces a discrepancy between $g(\zeta)$ and $\nabla_\zeta J$. The second component, $\frac{\alpha}{2}\nabla_\zeta\mathcal{R}(\zeta)$, is unaffected by this distribution mismatch because $\mathcal{R}$ is defined over the offline dataset. Together, these components yield

$$g_\mathcal{L}(\zeta) = g(\zeta) + \tfrac{\alpha}{2}\nabla_\zeta\mathcal{R}(\zeta). \tag{3.6}$$

In general, $g_\mathcal{L}$ is not the gradient of the regularized offline surrogate $J_\mathcal{D} + \frac{\alpha}{2}\mathcal{R}$ either. Differentiating equation 3.3 yields $\nabla_\zeta J_\mathcal{D} = g(\zeta) + \mathbb{E}_{\delta,x\sim d_\mathcal{D},u\sim\pi_\zeta(\cdot|x)}\big[\nabla_\zeta Q_{\pi_\zeta}(x,u)\big]$, where the second term captures the dependence of the critic on the policy and generally does not vanish. Holding the critic fixed during the actor update therefore gives a semi-gradient rather than the full gradient of $J_\mathcal{D}$ (Degris et al., 2012). Consequently, $g_\mathcal{L}$ need not be the gradient of any fixed scalar objective. Our analysis treats it as an inexact gradient of the deployment objective $\mathcal{L}$ and quantifies the resulting error floor in equation 4.10, with an explicit characterisation for LQR in Section 4.

Why this formulation. Unlike Park et al. (2025), our formulation uses a policy-gradient update that avoids differentiating Q. First, the average-cost objective measures steady-state performance, whereas discounting emphasises earlier costs (Puterman, 2014; Bertsekas, 2012). Second, stored (x,u) pairs can be relabelled with a new cost to retrain the actor and critic while reusing the flow model, without further environment interaction.

Algorithm 1 HeFLO — Heterogeneous FLow-regularized Offline policy learning

function $\mu_{\theta_b}(x, \eta)$
  for $t = 0, \dots, M-1$ **do** ▷ M Euler steps
    $\eta \leftarrow \eta + \frac{v_{\theta_b}(t/M, x, \eta)}{M}$
  end for
  return η
end function

▷ *Train Flow-policy* v_{θ_b}
while Not Converged **do**
  Sample a minibatch $(x, u) \sim \mathcal{D}$; $a^0 \sim \mathcal{N}(0, I_{n_u})$, $a^1 \leftarrow u$, $t \sim \mathrm{Unif}[0, 1]$
  $a^t \leftarrow (1-t)a^0 + ta^1$
  Update θ_b to minimize $\mathbb{E}\big[\|v_{\theta_b}(t, x, a^t) - (a^1 - a^0)\|_2^2\big]$ ▷ following equation 2.3
end while

▷ *Policy Learning (actor-critic)*
while Not Converged **do**
  Sample a minibatch $(x, u, c, x') \sim \mathcal{D}$
  ▷ *Train critic* Q_{π_ζ}: $\eta \sim \mathcal{N}(0, I_{n_u})$, $u' \leftarrow \pi_\zeta(x', \eta)$; update θ_q by TD learning (Tesauro, 1995)
  ▷ *Train actor* π_ζ: $\eta \sim \mathcal{N}(0, I_{n_u})$, $u \leftarrow \pi_\zeta(x, \eta)$
  Update $\zeta \leftarrow \zeta - \gamma\, g_{\mathcal{L}}$ ▷ $g_{\mathcal{L}}$ is defined in equation 3.6
end while

4 Theoretical Results

We analyse HeFLO for a family of heterogeneous LQR systems,

$$x_{t+1} = A(\delta)x_t + B(\delta)u_t + \omega_t, \quad c(x_t, u_t) = x_t^\top R_x x_t + u_t^\top R_u u_t, \tag{4.1}$$

with $R_x \succeq 0$, $R_u \succ 0$, and independent and identically distributed (i.i.d.) noise $\omega_t \sim \mathcal{N}(0, \Sigma_\omega(\delta))$. $A(\cdot)$, $B(\cdot)$, and $\Sigma_\omega(\cdot)$ are continuous on the compact set $\mathcal{S}_\delta$ with $\Sigma_\omega(\delta) \succ 0$. We denote $A_K(\delta) := A(\delta) - B(\delta)K$ and $\Sigma_h(\delta) := B(\delta)\Sigma_\eta B(\delta)^\top + \Sigma_\omega(\delta)$. The policy is assumed to be $\pi_K(\cdot|x) = \mathcal{N}(-Kx, \Sigma_\eta)$ and K ranges over the *uniformly* stabilizing set $\mathcal{K} := \{K : \rho(A_K(\delta)) < 1\ \forall \delta \in \mathcal{S}_\delta\}$. From this point onward we write K, which is a matrix, in place of ζ, which was assumed to be a vector. We restrict attention to linear state-feedback gains because, for a fixed δ, the optimal LQR policy is known to be linear in the state. From equation 4.1, the stationary state covariance is the solution of

$$\Sigma^K(\delta) = A_K(\delta)\Sigma^K(\delta)A_K(\delta)^\top + \Sigma_h(\delta), \qquad \bar{\Sigma}^K := \mathbb{E}_\delta[\Sigma^K(\delta)]. \tag{4.2}$$

The long-run average cost equation 3.2 then follows from the closed-form cost for a fixed δ (Yang et al., 2019) by taking the expectation over δ,

$$J(K) = \mathrm{Tr}\big[(R_x + K^\top R_u K)\bar{\Sigma}^K\big] + \mathrm{Tr}(R_u \Sigma_\eta), \tag{4.3}$$

while the offline surrogate equation 3.3 replaces $\bar{\Sigma}^K$ by the *empirical* moment $\Sigma_{\mathcal{D}} := \frac{1}{N_{\mathcal{D}}} \sum_i x_i x_i^\top$.

A fixed reference ball. Several constants below are suprema over a neighbourhood of $K = K_b$. Writing $\mathbb{B}(K_b, r) := \{K : \|K - K_b\|_F \le r\}$, we choose any r' with $\epsilon_b/\sqrt{\sigma_{\min}} < r' < r_b$ and set

$$\mathcal{B} := \mathbb{B}(K_b, r') \subset \mathcal{K}, \tag{4.4}$$

where $r_b := \sup\{r > 0 : \mathbb{B}(K_b, r) \subset \mathcal{K}\} > 0$ is the stability radius of the expert gain. All of L_J, G, ϵ_{cov}, ϵ_{het}, μ and L below are therefore suprema over this fixed $\mathcal{B}$.

4.1 Assumptions

Assumption 1 (Critic error, uniform on $\mathcal{B}$). *$\mathbb{P}\{\sup_{K\in\mathcal{B}} \|\hat{\Theta}_K - \Theta_K^{(m)}\|_F \le \epsilon_Q\} \ge 1 - p_Q$ for the TD estimator $\hat{\Theta}_K$, where $\Theta_K^{(m)}$ is given in equation 4.7 by replacing $\mathbb{E}_{\delta\sim\rho_\delta}$ with an empirical average $\frac{1}{m}\sum_{j=1}^m$ over m configurations in $\mathcal{D}$ (Proposition 1).*

Assumption 2 (Dataset coverage). *$\sigma_{\min} I \preceq \Sigma_{\mathcal{D}} \preceq \sigma_{\max} I$ with $\sigma_{\min} > 0$, where $\Sigma_{\mathcal{D}}$ is the realized empirical second moment of the dataset states.*

Assumption 3 (Flow fidelity and expert anchor). *Σ_η equals the action covariance of the demonstrators and the flow model is driven by the whitened noise $\Sigma_\eta^{-1/2}\eta$ (Section 3.2). There exists $K_b \in \mathcal{K}$ with $\mathbb{E}_{x\sim d_{\mathcal{D}},\eta}\|\mu_{\theta_b}(x,\eta) - (-K_b x + \eta)\|_2^2 \le \epsilon_b^2$.*

Assumption 1 requires the critic error to be uniformly bounded over $\mathcal{B}$ with high probability, ensuring that the bound applies to the data-dependent policy iterates. Assumption 2 is the matrix form of concentrability (Munos & Szepesvári, 2008; Chen & Jiang, 2019), whereas the condition $\sigma_{\min} > 0$ says that the demonstrations excite every state direction, without which no offline method can identify the gradient in the unexcited directions. Assumption 3 says only that the flow model has learned some stabilizing behaviour to within ϵ_b.

4.2 The algorithm's direction versus the gradient of the deployment cost

Proposition 1 (Structure of the value functions). *Let $K \in \mathcal{K}$. For each δ, $V_K^\delta(x) = x^\top P_K(\delta) x + v_c(\delta)$ with $P_K(\delta) \succeq 0$ the unique solution to*

$$P_K(\delta) = R_x + K^\top R_u K + A_K(\delta)^\top P_K(\delta) A_K(\delta). \tag{4.5}$$

Therefore, the aggregated action-value function $Q_K(x,u) := c(x,u) - J(K) + \mathbb{E}_{\delta,\omega}\big[V_K^\delta(A(\delta)x + B(\delta)u + \omega)\big]$ can be written in the following matrix form,

$$Q_K(x,u) = \begin{bmatrix} x \\ u \end{bmatrix}^\top \begin{bmatrix} \Theta_K^{xx} & \Theta_K^{xu} \\ \Theta_K^{ux} & \Theta_K^{uu} \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} + q_c, \text{ where} \tag{4.6}$$

$$\Theta_K^{xx} = R_x + \mathbb{E}_\delta[A^\top P_K A], \Theta_K^{xu} = (\Theta_K^{ux})^\top = \mathbb{E}_\delta[A^\top P_K B], \Theta_K^{uu} = R_u + \mathbb{E}_\delta[B^\top P_K B] \succeq R_u \succ 0, \tag{4.7}$$

and $q_c := \mathbb{E}_{\delta,\omega}[\omega^\top P_K(\delta)\omega] + \mathbb{E}_\delta[v_c(\delta)] - J(K)$ is finite. The argument δ of A, B and P_K is suppressed for notational convenience. Furthermore, $K \mapsto P_K(\delta)$ is real-analytic on $\mathcal{K}$ uniformly in δ, hence so are Θ_K, $\bar{\Sigma}^K$, J and $\mathcal{L}$.

The proof is in Appendix F.2. Note that the matrix Θ_K does not depend on the state or the action. We next define the per-configuration policy-gradient matrix $E_K(\delta) := \Theta_K^{uu}(\delta)K - B(\delta)^\top P_K(\delta) A(\delta)$ and its average $E_K := \mathbb{E}_\delta[E_K(\delta)]$, where $\Theta_K^{uu}(\delta)$ is defined as Θ_K^{uu} but for a fixed δ. Using equation 4.7 we can write

$$E_K = \Theta_K^{uu} K - \Theta_K^{ux}. \tag{4.8}$$

The central structural fact (Lemma 2, Appendix F.3) is that the true gradient and the computed direction *share the same policy-gradient matrix* and differ only in the covariance weighting it, i.e., $\nabla_K J(K) = 2\mathbb{E}_\delta[E_K(\delta)\Sigma^K(\delta)]$ while $g(K) = 2E_K\Sigma_{\mathcal{D}}$, so

$$\nabla_K J(K) - g(K) = \underbrace{2E_K(\bar{\Sigma}^K - \Sigma_{\mathcal{D}})}_{\text{distribution shift}} + \underbrace{2\Xi_K}_{\text{heterogeneity coupling}}, \Xi_K := \mathbb{E}_\delta[(E_K(\delta) - E_K)(\Sigma^K(\delta) - \bar{\Sigma}^K)].$$

We define $\epsilon_{\text{cov}} := \sup_{K\in\mathcal{B}} \|\bar{\Sigma}^K - \Sigma_{\mathcal{D}}\|_2$ and $\epsilon_{\text{het}} := \sup_{K\in\mathcal{B}} \|\Xi_K\|_F$. The first is ordinary distribution shift, which we retain explicitly in the analysis. Theorem 1(i) keeps each iterate K_k within distance $r(\alpha)$ of the behavioural anchor K_b. Consequently, Corollary 1 gives $\big\|\bar{\Sigma}^{K_k} - \Sigma_{\mathcal{D}}\big\|_2 \le L_\Sigma r(\alpha) + \epsilon_{\mathcal{D}}$, where L_Σ is the Lipschitz constant of $K \mapsto \bar{\Sigma}^K$ on $\mathcal{B}$ and $\epsilon_{\mathcal{D}} = \big\|\bar{\Sigma}^{K_b} - \Sigma_{\mathcal{D}}\big\|_2$ is the residual mismatch at the anchor. This bound applies along the iterates over the full reference ball $\mathcal{B}$, and the corresponding bound is $\epsilon_{\text{cov}} \le L_\Sigma r' + \epsilon_{\mathcal{D}}$. The second term, ϵ_{het}, is specific to heterogeneity and vanishes when $\mathcal{S}_\delta$ collapses to a point. The expert-aggregation gap $\epsilon_{\mathcal{D}}$ arises because Algorithm 2 collects each d_j under a per-configuration expert, so $\Sigma_{\mathcal{D}} \neq \bar{\Sigma}^{K_b}$ unless the experts agree.

4.3 Main result

Theorem 1 (Stability and convergence of HeFLO). *Consider equation 4.1 under Assumptions 1–3, let $K_0 \in \mathcal{K}$, let $\mathcal{B}, r'$ be as in equation 4.4, and let the update direction be computed from $\mathcal{D}$, whose $N_{\mathcal{D}}$ transitions come from m configurations. Let $D' := (\sqrt{\sigma_{\min}} r' - \epsilon_b)^2$ and $\Delta_{\mathcal{L}} := 4\varepsilon_{\text{tot}}^2/\mu$ and suppose*

$$\epsilon_b < \sqrt{\sigma_{\min}}\, r', \quad \mathcal{R}(K_0) < D', \quad \alpha > \max\Big\{ \tfrac{2(J(K_0)+\Delta_{\mathcal{L}})}{D'-\mathcal{R}(K_0)},\ \tfrac{2L_J}{\sigma_{\min}} \Big\}. \tag{4.9}$$

Then for any constant $\gamma \le 1/(2L)$, the iterates of Algorithm 1 satisfy, on a single event of probability at least $1-p_K$ simultaneously for all k: (i) Stability during learning. *Every iterate stabilizes every configuration, $K_k \in \mathcal{K}$, with $\|K_k - K_b\|_F \le r(\alpha) := (\epsilon_b + \sqrt{2(J(K_0)+\Delta_{\mathcal{L}})/\alpha + \mathcal{R}(K_0)})/\sqrt{\sigma_{\min}} \le r' < r_b$.* (ii) Convergence to a neighbourhood as

$$\mathcal{L}(K_k) - \mathcal{L}(K^\star) \le \big(1 - \tfrac{\gamma\mu}{2}\big)^k \big(\mathcal{L}(K_0) - \mathcal{L}(K^\star)\big) + \frac{3\varepsilon_{\text{tot}}^2}{\mu}, \tag{4.10}$$

where $K^\star$ minimizes $\mathcal{L}$ over $\mathcal{B}$ and the floor collects four sources,

$$\varepsilon_{\text{tot}} = \underbrace{2G\epsilon_{\text{cov}} + 2\epsilon_{\text{het}}}_{\text{offline + heterogeneity bias}} + \underbrace{c_1\epsilon_Q + c_4\epsilon_m}_{\text{critic, finite-}m}, \tag{4.11}$$

with $\epsilon_m \to 0$ as $m \to \infty$. Here $L := L_J + \alpha\sigma_{\max}$ and $\mu := \alpha\sigma_{\min} - L_J > 0$, with $L_J := \sup_{K\in\mathcal{B}} \|\nabla_K^2 J(K)\|_2 < \infty$. G, c_1, c_4 and $p_K = p_Q + p_m$ are given explicitly in Appendix F. Crucially, c_1 and c_4 do not depend on α.

The proof of Theorem 1 is provided in Appendix F.

Remark 1 (Behaviour-cloning warm start). The stability condition requires both sufficiently large α and a suitable initialization. Note that increasing α cannot compensate for $\mathcal{R}(K_0) \ge D'$. Initializing K_0 by minimizing $\mathcal{R}$ alone requires neither a critic nor environment interaction and gives $\mathcal{R}(K_0) \le \epsilon_b^2$. The initialization requirement is then satisfied whenever $\epsilon_b < \frac{1}{2}\sqrt{\sigma_{\min}}\, r'$, yielding $r(\alpha) = O(\epsilon_b + \alpha^{-1/2})$. Thus, a behaviour-cloning warm start provides a sufficient initialization when the flow model is accurate enough relative to the stability radius.

Remark 2 (Confinement under inexact updates). Inexact updates may increase $\mathcal{L}$ near the error floor, so confinement cannot rely on monotone descent. Appendix F.4 proves by induction that every iterate remains in the sublevel set $\{K : \mathcal{L}(K) \le \mathcal{L}(K_0) + \Delta_{\mathcal{L}}\}$. The slack $\Delta_{\mathcal{L}}$ in equation 4.9 accounts for these possible increases and vanishes with the error floor.

Remark 3 (Effect of heterogeneity). A wider heterogeneity range $\mathcal{S}_\delta$ covers more system configurations but can make uniform stabilization harder and increase the convergence error floor. Under the theorem's conditions, convergence remains geometric, although its rate and attainable accuracy depend on the system family. Appendix F.8 discusses these effects in detail.

Remark 4 (Step size and convergence rate). Stronger regularization requires a smaller step size, so the guaranteed convergence rate eventually saturates at a level determined by the dataset covariance conditioning. Appendix F.7 provides the full discussion.

Proposition 2 (The price of the regularizer)**.** *Suppose J attains its minimum over $\mathcal{K}$ at some $K_J^\star \in \mathcal{B}$, let $\alpha > L_J/\sigma_{\min}$, and put $c_x := \sqrt{\operatorname{Tr}\Sigma_{\mathcal{D}}}$. Then*

$$\|K^\star - K_J^\star\|_F \le \tfrac{\alpha}{\mu}\big(\sigma_{\max}\|K_J^\star - K_b\|_F + \epsilon_b c_x\big), \quad J(K^\star) - J(K_J^\star) \le \tfrac{L_J}{2}\|K^\star - K_J^\star\|_F^2. \tag{4.12}$$

The right-hand side stays bounded as $\alpha \to \infty$, with limit $\frac{L_J}{2}(\frac{\sigma_{\max}}{\sigma_{\min}}\|K_J^\star - K_b\|_F + \frac{c_x}{\sigma_{\min}}\epsilon_b)^2$, and vanishes when the demonstrations are optimal and the flow model is exact.

A proof of the proposition is given in Appendix F.6.

Reading the two parts of Theorem 1. Part (i) does not follow from any property of J itself. An offline algorithm cannot evaluate J, and every dataset quantity stays finite as $K \to \partial\mathcal{K}$, so no monotonicity of computed quantities can certify stability. The regulariser gives the behaviour-cloning term a role beyond variance reduction. This also addresses a difficulty generic to domain randomization for control: since $\mathcal{K}$ is an intersection over $\mathcal{S}_\delta$, a step on the randomized objective can destabilize some configurations even while the average improves, which Fujinami et al. (2025) handle by masking gradients of unstabilized scenarios, without a convergence guarantee. An expert anchor removes the need for masking, and is available only because the setting is offline and demonstration-driven. Part (ii) decays to a floor and no further: a fixed ϵ_Q, a fixed $N_{\mathcal{D}}$, a fixed m and a fixed dataset produce a persistent bias, and inexact gradient descent under bias reaches a ball proportional to it. Therefore, exact convergence needs all four sources to vanish jointly, which is the classical on-policy LQR setting (Fazel et al., 2018). Two of these terms, $\epsilon_{\mathcal{D}}$ and ϵ_m, can be reduced only by using more varied configurations, rather than by adjusting α, training longer, or collecting more transitions from the same experts. The deployment-cost bias is controlled by the suboptimality of the demonstrations and is not increasing in α, so α may be raised to buy stability without paying for it beyond a fixed, expert-determined amount.

4.4 From the LQR analysis to the nonlinear experiments

Our guarantees apply to linear systems with quadratic costs, while Section 5 evaluates HeFLO on nonlinear MuJoCo environments. The theory provides three qualitative insights for these experiments. First, using dataset states instead of policy-induced states introduces a bias through policy–data mismatch, as Lemma 2 makes explicit. The same issue arises in nonlinear systems. Second, behavioural regularisation keeps the policy close to the demonstrations, which can help limit distribution shift and critic error. This motivates testing how performance depends on α, although the stability threshold is established only for LQR. Third, a wider range $\mathcal{S}_\delta$ covers more configurations but can increase differences between their policy gradients, making a single policy harder to learn (Tobin et al., 2017). In our LQR analysis, this coupling is quantified by ϵ_{het}.

The formal guarantees do not carry over directly. The proof of equation 4.10 relies on gradient dominance of the regularised objective, the quadratic structure of Q_K, and a stability neighbourhood around K_b. These properties are not established for the neural policies used in our experiments, so we make no claim of provable stability or convergence there. LQR nevertheless provides a tractable setting in which these mechanisms can be analysed rigorously (Fazel et al., 2018; Hu et al., 2023; Yang et al., 2019), while the MuJoCo experiments assess their practical relevance beyond linear systems.

4.5 Limitations

(i) *Policy class:* the theory covers only linear state-feedback gains. Since δ is unobserved at deployment, using the state history may improve performance (Chen et al., 2022). We do not bound this additional gap or cover the neural policies used in the experiments. (ii) *Critic accuracy:* Assumption 1 is not proved. The aggregate Q_K averages over configurations and need not satisfy a single Bellman equation, so pooled TD learning need not recover this target. This is one of our future research directions. (iii) *Coverage:* $\sigma_{\min} > 0$ is required, that is, the data must excite every state direction. (iv) *Residual error:* the bound in equation 4.10 generally has a nonzero floor for a fixed dataset and critic. (v) *Regularisation strength:* the stability threshold depends on the unknown radius r_b, so it motivates a warm start but does not provide a directly usable value of α. (vi) *Role of flow matching:* the guarantees rely on an accurate behavioural anchor rather than on any property specific to flow matching; they hold verbatim for a mean-squared-error regulariser of comparable fidelity, and an empirical ablation separating the two is left to future work.

5 Experimental Analysis

In this section, we present the empirical results of HeFLO across heterogeneous continuous control environments. First, we apply HeFLO to a LQR setup, a linearised inverted pendulum (see Appendix D.1 for the LQR setup results), and then experiment with several nonlinear systems from MuJoCo (Todorov et al., 2012) available in Gymnasium (Towers et al., 2024). Specifically, we evaluate on seven environments: `InvertedPendulum-v5`, `InvertedDoublePendulum-v5`, `Pusher-v5`, `Reacher-v5`, `Swimmer-v5`, `Hopper-v5`, and `Walker2d-v5`, which range from simple to complex systems. One of the datasets for testing offline RL algorithms is OGBench, but its existing datasets do not directly cover our proposed heterogeneous systems setting. Though the different systems available in OGBench can also be modified to generate such heterogeneous data, but this would constitute a custom extension rather than the standard benchmark.

Offline Expert Data Generation. For each environment, we construct a family of heterogeneous instances by sampling system parameters uniformly from a prescribed range. A separate expert policy is then obtained for each instance by training a SAC (Haarnoja et al., 2018) or PPO (Schulman et al., 2017) agent to convergence. Each expert policy is subsequently used to collect trajectories on its corresponding environment instance, which are aggregated to form the offline dataset $\mathcal{D}$.

Architecture Design and Hyperparameters. The flow model, actor, and critic each use a 4-layer neural network with hidden dimensions [512, 512, 512, 512]. We use `ReLU` activations in all layers except the last one. We train all models with `Adam` optimizer using a learning rate of 3×10^{-4}. Further implementation details are given in Appendix E.

Training Process. HeFLO training proceeds in two stages: *Stage 1*, behaviour flow learning, in which the behaviour policy is trained to capture the expert data distribution; and *Stage 2*, one-step policy learning, in which the pre-trained behaviour policy is used to compute the behaviour- cloning

regularisation term. Training runs for several thousand epochs, and policy performance is evaluated periodically by computing the mean episodic reward over multiple evaluation episodes.

Result Discussion. Table 1 compares the proposed HeFLO method with existing offline algorithms. We use the average normalised score (reward; i.e. negative of cost in our problem formulation) as the performance measure. For each environment, we run every algorithm for 20 episodes using 5 different seeds, then take the mean to get the average reward. We normalise this average reward (Fu et al., 2020) to account for the different reward scales across environments. The highest score in each row is highlighted, and $\pm$ denotes the standard deviation across seeds. Figure 2 in Appendix D.2 shows the convergence of HeFLO. Across all environments, HeFLO reaches near-expert performance within a few thousand epochs and remains stable thereafter. Its standard deviation across seeds is also the smallest of the four methods in six of the seven environments, the exception being `Reacher-v5`, where CQL is tighter. HeFLO therefore produces stable, low-variance policies across trials.

Table 1: Average normalised score of HeFLO and existing offline RL algorithms. The highest score in each row is highlighted; $\pm$ denotes the standard deviation across five seeds.

Environment	IQL	CQL	TD3+BC	HeFLO
InvertedPendulum-v5	57.06 $\pm$ 14.60	100 $\pm$ 0.0	100 $\pm$ 0.0	100 $\pm$ 0.0
Pusher-v5	98.24 $\pm$ 5.57	106.54 $\pm$ 3.99	103.58 $\pm$ 3.85	104.70 $\pm$ 3.81
InvertedDoublePendulum-v5	94.01 $\pm$ 17.27	78.69 $\pm$ 34.07	102.64 $\pm$ 0.02	102.63 $\pm$ 0.0
Reacher-v5	98.25 $\pm$ 6.57	103.06 $\pm$ 3.37	97.72 $\pm$ 4.44	100.17 $\pm$ 4.73
Swimmer-v5	105.54 $\pm$ 1.49	100.35 $\pm$ 3.17	104.22 $\pm$ 0.39	103.89 $\pm$ 0.23
Hopper-v5	115.22 $\pm$ 4.17	112.42 $\pm$ 7.68	117.64 $\pm$ 0.62	119.05 $\pm$ 0.07
Walker2d-v5	91.12 $\pm$ 18.76	93.02 $\pm$ 16.03	102.86 $\pm$ 12.47	109.54 $\pm$ 2.66

Reward restructuring. As discussed in Section 3.2, we further analyse the effect of modifying the reward function on the learned policy in the `Hopper-v5` environment. The results (discussed in detail in Appendix D.5) show that an offline policy can be learned from a modified reward (cost) structure that differs from the one the experts optimised. This illustrates a practical advantage of the HeFLO framework, i.e., the reward function can be changed without additional environment interaction or data collection.

6 Conclusion

We introduced HeFLO for offline policy learning across heterogeneous systems using an infinite-horizon average-cost objective and flow-based behavioural regularisation. For heterogeneous LQR, under the stated assumptions and initialization conditions, sufficiently strong regularisation ensures that every policy iterate stabilises every configuration. The regularised objective gap decreases geometrically to an error floor that explicitly accounts for distribution mismatch, heterogeneity, and estimation errors. In particular, additional transitions from the same experts remove neither the expert-aggregation gap nor the configuration-sampling error. The regularisation bias is controlled by demonstration quality and flow-model accuracy and remains bounded as α grows. Empirically, HeFLO almost matches or exceeds IQL, CQL and TD3+BC on seven heterogeneous MuJoCo families while producing markedly lower seed-to-seed variance. However, the comparison is restricted to general-purpose offline RL baselines, and dedicated robust or domain-randomisation methods remain to be evaluated. Future work includes a joint actor–critic analysis relaxing Assumption 1, policies that infer δ from observed histories, comparisons against robust offline RL baselines, and federated learning from distributed expert datasets.

AI USE STATEMENT

In this work, we utilised generative AI tools primarily for drafting and editing text, as well as suggesting the manuscript structure. Additionally, we received assistance with coding the proposed algorithm and baseline algorithms. We also sought help in formulating mathematical claims, drafting proofs, reviewing research methodology and experiments, and interpreting results. Specifically, two large language models helped critically review the manuscript: one provided by the conference organisers and another obtained through a paid subscription.

We have not used generative AI tools to generate synthetic datasets, develop theoretical models or conceptual frameworks, or propose or refine research hypotheses. Tasks such as translation, dataset cleaning and reformatting, and qualitative or thematic data analysis are not applicable to this work.

All theoretical statements, proofs, algorithmic contributions, and experiments are the work of the authors, who have verified every derivation presented here and take full responsibility for the content of the paper.

ETHICS STATEMENT

This work does not involve human subjects, personal data, or sensitive information. We use publicly available datasets and standard reinforcement learning environments for our experiments. The methods developed in this work are intended for research purposes, and we are not aware of any specific ethical concerns or potential harms arising from the methodology or experimental results presented in this paper. We have followed the relevant dataset and software licenses and have made efforts to accurately report our experimental setup and results.

REPRODUCIBILITY STATEMENT

The complete implementation of HeFLO, the offline data-generation scripts, the expert checkpoints, and the scripts for reproducing every table and figure are provided as a zip file in the supplementary materials. We also provide all the relevant information for reproducing the algorithm and its results throughout the paper. Appendix C specifies the data-generation protocol, Appendix E presents the architectures, hyperparameters, and software versions, and Appendix D.6 states the compute used. Proofs of every formal statement are given in full in Appendix F, with the assumptions collected in Section 4.1. The code necessary to reproduce the results is included as supplementary material with the manuscript submission.

REFERENCES

Rishabh Agarwal, Dale Schuurmans, and Mohammad Norouzi. An optimistic perspective on offline reinforcement learning. In *International Conference on Machine Learning*, pp. 104–114. PMLR, 2020.

Anurag Ajay, Yilun Du, Abhi Gupta, Joshua B Tenenbaum, Tommi S Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision-making? In *International Conference on Learning Representations (ICLR)*, 2023.

Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 34(6):26–38, 2017.

Christopher G Atkeson and Stefan Schaal. Robot learning from demonstration. In *International Conference on Machine Learning*, volume 97, pp. 12–20, 1997.

Jonathan Baxter and Peter L Bartlett. Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research*, 15:319–350, 2001.

Felix Berkenkamp, Matteo Turchetta, Angela Schoellig, and Andreas Krause. Safe model-based reinforcement learning with stability guarantees. *Advances in Neural Information Processing Systems*, 30, 2017.

Dimitri P Bertsekas. *Dynamic Programming and Optimal Control, Vol. II: Approximate Dynamic Programming*. Athena Scientific, 4th edition, 2012.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

Jinglin Chen and Nan Jiang. Information-theoretic considerations in batch reinforcement learning. In *International Conference on Machine Learning*, pp. 1042–1051. PMLR, 2019.

Xiaoyu Chen, Jiachen Hu, Chi Jin, Lihong Li, and Liwei Wang. Understanding domain randomization for sim-to-real transfer. In *International Conference on Learning Representations (ICLR)*, 2022.

OX-Embodiment Collaboration, Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 1(2), 2023.

Antonio Coronato, Muddasar Naeem, Giuseppe De Pietro, and Giovanni Paragliola. Reinforcement learning for intelligent healthcare applications: A survey. *Artificial Intelligence in Medicine*, 109: 101964, 2020.

Sarah Dean, Horia Mania, Nikolai Matni, Benjamin Recht, and Stephen Tu. On the sample complexity of the linear quadratic regulator. *Foundations of Computational Mathematics*, 20(4):633–679, 2020.

Thomas Degris, Martha White, and Richard S. Sutton. Off-policy actor-critic. In *International Conference on Machine Learning*, 2012.

Pierluca D'Oro and Wojciech Jaśkowski. How to learn a useful critic? model-based action-gradient-estimator policy optimization. *Advances in Neural Information Processing Systems*, 33:313–324, 2020.

Finale Doshi-Velez and George Konidaris. Hidden parameter Markov decision processes: A semi-parametric regression approach for discovering latent task parametrizations. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2016.

Hidayet Ersin Dursun, Yusuf Güven, and Tufan Kumbasar. Imitation learning for autonomous driving: insights from real-world testing. In *2025 7th International Congress on Human-Computer Interaction, Optimization and Robotic Applications (ICHORA)*, pp. 1–6. IEEE, 2025.

Maryam Fazel, Rong Ge, Sham Kakade, and Mehran Mesbahi. Global convergence of policy gradient methods for the linear quadratic regulator. In *International Conference on Machine Learning*, pp. 1467–1476. PMLR, 2018.

Pete Florence, Corey Lynch, Andy Zeng, Oscar A Ramirez, Ayzaan Wahid, Laura Downs, Adrian Wong, Johnny Lee, Igor Mordatch, and Jonathan Tompson. Implicit behavioral cloning. In *Conference on Robot Learning*, pp. 158–168. PMLR, 2022.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4RL: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

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

Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, pp. 1587–1596. PMLR, 2018.

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International Conference on Machine Learning*, pp. 2052–2062. PMLR, 2019.

Tesshu Fujinami, Bruce D Lee, Nikolai Matni, and George J Pappas. Domain randomization is sample efficient for linear quadratic control. In *Proceedings of the 7th Annual Learning for Dynamics and Control Conference*, volume 283 of *Proceedings of Machine Learning Research*, pp. 907–919. PMLR, 2025.

Javier Garcıa and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(1):1437–1480, 2015.

Vinicius G Goecks, Gregory M Gremillion, Vernon J Lawhern, John Valasek, and Nicholas R Waytowich. Integrating behavior cloning and reinforcement learning for improved performance in dense and sparse reward environments. *arXiv preprint arXiv:1910.04281*, 2019.

Benjamin Gravell, Peyman Mohajerin Esfahani, and Tyler Summers. Learning optimal controllers for linear systems with multiplicative noise via policy gradient. *IEEE Transactions on Automatic Control*, 66(11):5283–5298, 2021.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pp. 1861–1870. Pmlr, 2018.

Philippe Hansen-Estruch, Ilya Kostrikov, Michael Janner, Jakub Grudzien Kuba, and Sergey Levine. IDQL: Implicit Q-learning as an actor-critic method with diffusion policies. *arXiv preprint arXiv:2304.10573*, 2023.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *Advances in Neural Information Processing Systems*, 29, 2016.

Bin Hu, Kaiqing Zhang, Na Li, Mehran Mesbahi, Maryam Fazel, and Tamer Başar. Toward a theoretical foundation of policy optimization for learning control policies. *Annual Review of Control, Robotics, and Autonomous Systems*, 6(1):123–158, 2023.

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

B Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A Al Sallab, Senthil Yogamani, and Patrick Pérez. Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(6):4909–4926, 2021.

Bahare Kiumarsi, Kyriakos G Vamvoudakis, Hamidreza Modares, and Frank L Lewis. Optimal and autonomous control using reinforcement learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 29(6):2042–2062, 2017.

Vijay Konda and John Tsitsiklis. Actor-critic algorithms. *Advances in Neural Information Processing Systems*, 12, 1999.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*, 2021.

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.

Sascha Lange, Thomas Gabel, and Martin Riedmiller. Batch reinforcement learning. In *Reinforcement Learning: State of the Art*, pp. 45–73. Springer, 2012.

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.

Lanqing Li, Rui Yang, and Dijun Luo. FOCAL: Efficient fully-offline meta-reinforcement learning via distance metric learning and behavior regularization. In *International Conference on Learning Representations (ICLR)*, 2021.

Yuxi Li. Deep reinforcement learning: An overview. *arXiv preprint arXiv:1701.07274*, 2017.

Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky TQ Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code. *arXiv preprint arXiv:2412.06264*, 2024.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. *arXiv preprint arXiv:2108.03298*, 2021.

Eric Mitchell, Rafael Rafailov, Xue Bin Peng, Sergey Levine, and Chelsea Finn. Offline meta-reinforcement learning with advantage weighting. In *International Conference on Machine Learning (ICML)*, 2021.

Rémi Munos and Csaba Szepesvári. Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research*, 9(27):815–857, 2008.

Ashvin Nair, Bob McGrew, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Overcoming exploration in reinforcement learning with demonstrations. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6292–6299. IEEE, 2018.

Andrew Y Ng and Stuart J Russell. Algorithms for inverse reinforcement learning. In *International Conference on Machine Learning (ICML)*, volume 1, pp. 2, 2000.

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

Kishan Panaganti, Zaiyan Xu, Dileep Kalathil, and Mohammad Ghavamzadeh. Robust reinforcement learning using offline data. *Advances in Neural Information Processing Systems*, 35:80494–80507, 2022.

Seohong Park, Qiyang Li, and Sergey Levine. Flow q-learning. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=KVf2SFL1pi.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32, 2019.

Fernando Pérez-Cruz. Kullback-leibler divergence estimation of continuous distributions. In *IEEE International Symposium on Information Theory*, pp. 1666–1670. IEEE, 2008.

Rafael Figueiredo Prudencio, Marcos ROA Maximo, and Esther Luna Colombini. A survey on offline reinforcement learning: Taxonomy, review, and open problems. *IEEE Transactions on Neural Networks and Learning Systems*, 35(8):10237–10257, 2023.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

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. *arXiv preprint arXiv:1709.10087*, 2017.

Benjamin Recht. A tour of reinforcement learning: The view from continuous control. *Annual Review of Control, Robotics, and Autonomous Systems*, 2(1):253–279, 2019.

Ali Riahi Samani, Xujiang Zhao, and Feng Chen. Distribution shift, generalization and ood challenge in offline reinforcement learning: a comprehensive survey. *Neural Computing and Applications*, 38(6):150, 2026.

Georg Schildbach, Lorenzo Fagiano, Christoph Frei, and Manfred Morari. The scenario approach for stochastic model predictive control with bounds on closed-loop constraint violations. *Automatica*, 50(12):3009–3018, 2014.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, pp. 1889–1897. PMLR, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *International Conference on Machine Learning*, pp. 387–395. Pmlr, 2014.

Charles M Stein. Estimation of the mean of a multivariate normal distribution. *The Annals of Statistics*, pp. 1135–1151, 1981.

Richard S Sutton and Andrew G Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2 edition, 2018.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in Neural Information Processing Systems*, 12, 1999.

Denis Tarasov, Alexander Nikulin, Dmitry Akimov, Vladislav Kurenkov, and Sergey Kolesnikov. CORL: Research-oriented deep offline reinforcement learning library. In *3rd Offline RL Workshop: Offline RL as a "Launchpad"*, 2022. URL https://openreview.net/forum?id=SyAS49bBcv.

Gerald Tesauro. Temporal difference learning and td-gammon. *Communications of the ACM*, 38(3): 58–68, 1995.

Gabriele Tiboni, Karol Arndt, and Ville Kyrki. DROPO: Sim-to-real transfer with offline domain randomization. *Robotics and Autonomous Systems*, 166:104452, 2023.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 23–30. IEEE, 2017.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012. doi: 10.1109/IROS.2012.6386109.

Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. *arXiv preprint arXiv:1805.01954*, 2018.

Leonardo F Toso, Donglin Wang, and James Anderson. Meta-learning linear quadratic regulators: A policy gradient MAML approach for model-free LQR. *arXiv preprint arXiv:2401.14534*, 2024.

Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, et al. Gymnasium: A standard interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032*, 2024.

He Wang, Laixi Shi, and Yuejie Chi. Sample complexity of offline distributionally robust linear markov decision processes. *arXiv preprint arXiv:2403.12946*, 2024.

Zhendong Wang, Jonathan J Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2023.

Thomas Wolf, Lysandre Debut, Victor Sanh, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

Farnaz Adib Yaghmaie and Arunava Naha. Convergence of flow-policy gradient learning for linear quadratic regulator problems. *arXiv preprint arXiv:2511.11131*, 2025.

Zhuoran Yang, Yongxin Chen, Mingyi Hong, and Zhaoran Wang. Provably global convergence of actor-critic: A case for linear quadratic regulator with ergodic cost. *Advances in Neural Information Processing Systems*, 32, 2019.

Kaiqing Zhang, Bin Hu, and Tamer Başar. Policy optimization for $\mathcal{H}_2$ linear control with $\mathcal{H}_\infty$ robustness guarantee: Implicit regularization and global convergence. *SIAM Journal on Control and Optimization*, 59(6):4081–4109, 2021.

A Background

Actor–critic methods. AC methods combine an *actor* representing the policy $\pi(\cdot|x)$ with a *critic* evaluating it through a value $V_\pi(x)$ or action-value $Q_\pi(x,u)$. For continuous state-action MDPs both are approximated by function approximators, linear or neural, and trained iteratively (Arulkumaran et al., 2017; Li, 2017). The critic may be trained by temporal-difference learning, gradient TD, Monte Carlo or least-squares methods (Sutton & Barto, 2018). Under the average-cost setting used in our study, the relevant recursion is the relative TD error of Algorithm 1, which requires a running estimate of the average cost J. The actor is updated by policy gradient (Sutton et al., 1999; Silver et al., 2014). Decoupling the two stabilizes optimization by separating policy improvement from critic estimation.

Offline RL. Offline RL (Levine et al., 2020; Prudencio et al., 2023; Agarwal et al., 2020) learns the parameter ζ of a policy π_ζ minimizing a long-term objective from a fixed dataset $\mathcal{D} = \{d_j\}$, each d_j a set of transitions (x_t, u_t, c_t, x_{t+1}), without further environment interaction. The data may come from any behaviour policy. This enables application where online interaction is costly or unsafe, as in autonomous driving (Dursun et al., 2025) and healthcare (Coronato et al., 2020).

Flow-based behaviour cloning. Behaviour cloning (Florence et al., 2022) learns to imitate actions in an offline dataset and is often used to regularise policy learning, balancing cost minimisation with proximity to the demonstrated behaviour. Flow matching (Lipman et al., 2024; Liu et al., 2022) models complex data distributions by matching the flow of points from a known input distribution to an unknown target. A parameterized vector field $v_{\theta_b} : [0,1] \times \mathcal{X} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_u}$ is trained by equation 2.3, in which $a^t = (1-t)a^0 + ta^1$ interpolates a noise sample $a^0 \sim \mathcal{N}(0, I_{n_u})$ and a demonstrated action $a^1 = u$. The field generates a state-dependent flow $\psi_{\theta_b}(t, x, a)$, and we write $\mu_{\theta_b}(x, z) := \psi_{\theta_b}(1, x, z)$ for its time-1 map, which acts as a cloned policy transporting noise z into an action. Recovering μ_{θ_b} requires solving an ODE, which is prohibitive at every control step. Therefore, following Park et al. (2025), we use the flow policy as a regularizer rather than deploying it.

B Extended related work

Offline RL. Offline RL (Lange et al., 2012; Levine et al., 2020) learns policies entirely from pre-collected datasets without further interaction. The fundamental challenge is overestimation of the Q-function under distributional shift between the learned policy and the data (Fujimoto et al., 2019). CQL (Kumar et al., 2020) penalizes Q-values at out-of-distribution actions; IQL (Kostrikov et al., 2021) avoids querying them at all by estimating the value function via expectile regression.

Learning from demonstration. Common approaches include behaviour cloning (Torabi et al., 2018), inverse RL (Ng & Russell, 2000) and generative adversarial imitation learning (Ho & Ermon, 2016); one of the earliest works is that of Atkeson & Schaal (1997). Behaviour cloning has since been used as a regulariser in policy optimisation: Nair et al. (2018) and Goecks et al. (2019) combine it with DDPG, and Rajeswaran et al. (2017) with natural policy gradients, though these rely on replay-buffer modifications and pre-training; Fujimoto & Gu (2021) adds it directly to an existing policy-gradient method (Baxter & Bartlett, 2001; Konda & Tsitsiklis, 1999; Sutton et al., 1999). All of these clone the demonstrator's exact actions, which limits expressiveness as datasets (Collaboration et al., 2023) and their distributions grow more complex (Riahi Samani et al., 2026), motivating richer policy classes (Mandlekar et al., 2021). FQL (Park et al., 2025) integrates a flow-based generative model into an actor–critic framework for this purpose, and Yaghmaie & Naha (2025) analyse it under LQR.

Latent-parameter, contextual and meta offline RL. Because δ is unobserved at deployment, the family equation 3.1 is a hidden-parameter MDP (Doshi-Velez & Konidaris, 2016), and an alternative to a single context-free policy is to infer a task representation and condition on it, as in offline meta-RL (Li et al., 2021; Mitchell et al., 2021). We do not, for two reasons: a context encoder must be identified from the same fixed dataset, which reintroduces the coverage problem at a second level; and our deployment setting gives no dedicated adaptation phase, so a policy must act correctly from the first time step. The latter does not rule out history-conditioned policies, which can adapt implicitly within an episode; it means only that they pay an early-step suboptimality that a memoryless policy analysis does not have to model. We therefore restrict attention to memoryless policies, and a context-conditioned variant is the natural next step rather than a baseline we outperform.

Policy gradient for LQR under uncertainty. The closest theoretical results are policy-gradient analyses of LQR beyond the single nominal system: multiplicative-noise LQR (Gravell et al., 2021), robust and mixed-objective LQR policy optimization (Zhang et al., 2021), meta and multi-task LQR policy gradient (Toso et al., 2024), and the sample complexity of learning LQR from data (Dean et al., 2020). Relative to these, our setting is offline (no rollouts of the current policy at any point), the objective is average cost rather than discounted, and the regularizer is what supplies the stability certificate.

Robust MDPs and domain randomization. Robust MDPs (Iyengar, 2005; Nilim & El Ghaoui, 2005) optimize worst-case performance over an uncertainty set of transition kernels. Offline approaches include Robust Fitted Q-Iteration (Panaganti et al., 2022) and sample-complexity analysis for distributionally robust linear MDPs (Wang et al., 2024). Domain randomization (Tobin et al., 2017) instead optimizes average performance over a distribution of system parameters. DROPO (Tiboni et al., 2023) estimates this distribution from offline trajectories, without establishing policy convergence or closed-loop stability. HeFLO adopts the average-case objective but learns the policy entirely from fixed demonstrations, without observing configuration identifiers, and provides stability and convergence analysis for heterogeneous LQR.

Fujinami et al. (2025) study the linear quadratic form $K_{\mathrm{DR}}(\mathcal{D}) = \arg\min_K \mathbb{E}_{\theta\sim\mathcal{D}}[C(K,\theta)]$ with $\mathcal{D}$ a confidence ellipsoid around a least-squares estimate, and show that with an appropriately chosen sampling distribution domain randomization attains the optimal $1/N$ asymptotic decay of excess cost, matching certainty equivalence and improving on robust control, whose leading constant depends on an operator norm rather than a trace. This is a strong argument for the average-case objective we adopt and the reason we do not pursue a worst-case formulation. It differs from our setting in three respects that delimit our contribution: (i) theirs is model-based — the randomization distribution comes from system identification, and the controller is then synthesized — whereas ours is model-free and offline, with no identification step and no ability to query the system; (ii) their policy-gradient procedure masks the gradient contributions of scenarios the current iterate fails to stabilize and, as they state, lacks a formal convergence guarantee, whereas Theorem 1 supplies convergence together with a certificate that every iterate stabilises every configuration; and (iii) they analyse the population objective directly, without accounting for the error incurred when the expectation is approximated by finitely many sampled systems, which is exactly what ϵ_m measures.

Chen et al. (2022) analyse domain randomization for sim-to-real transfer by modelling the simulator as a set of MDPs with unobserved tunable parameters, give sharp bounds on the sim-to-real gap between the domain randomization policy and the optimal policy for the real system, and show that transfer can succeed under mild conditions with no real-world samples.

C Data Preparation

Our experimental setup requires offline data collected from a family of heterogeneous systems, where heterogeneity arises from variations in system parameters within a bounded uncertainty set. For each system configuration, a separate expert policy is trained on the corresponding parametric instance, and offline trajectories are collected using that expert. Algorithm 2 outlines the complete offline data-generation procedure. The number of episodes collected per expert, and the number of experts, vary across environments; Table 2 summarizes both.

Table 2: Expert-generated offline data details

Environment	Episodes per expert	# of Experts
Swimmer	250	4
Pusher	200	5
Reacher	200	5
InvertedPendulum	200	5
InvertedDoublePendulum	200	5
Hopper	200	5
Walker2d	200	5

Algorithm 2 Offline Data Preparation

Require: **Env**: Environment, $\mathbf{m}$: Number of heterogeneous systems, $\mathcal{S}_\delta$: support of δ, sampled as $\mathrm{Unif}(\delta_{\min}, \delta_{\max})$, e: number of episodes, s: number of steps per episode, $\mathcal{D}$: empty offline dataset
$i \leftarrow 1$, $episode \leftarrow 1$, $step \leftarrow 1$
while $i \leq \mathbf{m}$ **do**
  1. Choose δ from $\mathcal{S}_\delta$
  2. Create environment env_i from **Env** by setting its system parameters according to δ
  3. Train an expert policy π_{E_i} on env_i
  4. Start env_i from a random state x_0 and roll out π_{E_i}
  while $episode \leq e$ **do**
    while $step \leq s$ **do**
      5a. Generate the transition $(x_{step}, u_{step}, c_{step}, x'_{step})$
      5b. Store the transition in $\mathcal{D}$
      $step \leftarrow step + 1$
    end while
    $episode \leftarrow episode + 1, \quad step \leftarrow 1$
  end while
  $episode \leftarrow 1, \quad i \leftarrow i + 1$
end while

Expert Trajectory Generation. Expert policies are central to our offline data generation procedure. For each environment, we generate expert policies as follows. We first select an environment from Gymnasium (Towers et al., 2024) and sample its system parameters uniformly from a prescribed range, following the bounded heterogeneity assumption introduced in Section 3.1. The sampled parameter values are then used to instantiate the corresponding environment (Brockman et al., 2016; Towers et al., 2024). For each instantiated environment, we train either a Soft Actor-Critic (SAC) or a Proximal Policy Optimization (PPO) agent to convergence using existing training utilities from the Hugging Face (Wolf et al., 2019) library, yielding the expert policy for that parametric configuration. This procedure is repeated for each environment and each sampled parameter configuration. Table 3 summarizes the system parameters, their admissible ranges, and the training algorithm used for each environment. For most environments, we choose the range based on absolute parameter values. The two exceptions are Hopper-v5 and Walker2d-v5. For these, we choose a range of scaling factors so that each limb or body part scales by a similar amount, keeping the environment's original design intact.

D Discussion on Experimental Analysis

D.1 Simulation on Linear System (LQR setup)

We use a linearized inverted pendulum from gymnasium (classic control) (Towers et al., 2024) as a benchmark system for the simulation study. The system state consists of the pendulum angle and angular velocity, given by $x_k = [\theta_k,\ \dot{\theta}_k]^\top$, with the angular velocity constrained to $|\dot{\theta}_k| \leq 8\,\mathrm{rad/s}$. The applied torque is the control input and satisfies $|u_k| \leq 2\,\mathrm{N\,m}$. The process noise has covariance $\Sigma_\omega = 10^{-4} I$. The dynamics are obtained by linearizing the pendulum about its upright equilibrium. We evaluate the control cost in equation 4.1 using $R_x = \mathrm{diag}(1, 0.1)$ and $R_u = 0.001$. The system dynamics follow equation 4.1, and the resulting discrete-time system matrices are

$$A = \begin{bmatrix} 1 & \Delta t \\ -\frac{g}{l}\Delta t & 1 - \frac{b}{m_p l^2}\Delta t \end{bmatrix}, \qquad B = \begin{bmatrix} 0 \\ \frac{\Delta t}{m_p l^2} \end{bmatrix}, \tag{D.1}$$

where m_p, l, b, g and Δt denote the pendulum mass, rod length, viscous damping coefficient, gravitational acceleration and sampling interval, respectively (we write m_p for the mass to avoid a clash with the number of configurations m). Throughout the simulations we set $b = 0$, $g = 10\,\mathrm{m/s^2}$ and $\Delta t = 0.05\,\mathrm{s}$, while m_p and l are drawn at random from a prescribed range.

For the offline learning experiment, we first choose a pair of values of m_p and l to create a specific linearized pendulum system, and then generate state-action trajectories together with their rewards for 200 steps under the control of an expert controller. We repeat this trajectory-generation step for

Table 3: System parameters' range and the type of expert policy

Environment	System parameters	Range	Expert policy
Reacher	Length of first arm Length of second arm Density	$[0.09, 0.11]$ $[0.09, 0.11]$ $[800, 1200]$	SAC
Swimmer	Length of link 1 Viscosity Density of links	$[0.8, 1.2]$ $[0.09, 0.11]$ $[800, 1200]$	PPO
Pusher	Length of link 1 Length of link 2 Density	$[0.35, 0.45]$ $[0.271, 0.311]$ $[270, 330]$	SAC
InvertedPendulum	Length of pole Radius of pole Cart size	$[0.55, 0.65]$ $[0.047, 0.051]$ $[0.09, 0.11]$	SAC
InvertedDoublePendulum	Length of pole 1 Length of pole 2 Cart size	$[0.55, 0.65]$ $[0.55, 0.65]$ $[0.09, 0.11]$	SAC
Hopper	Limb length Mass	$[0.9, 1.1]$ $[0.95, 1.05]$	SAC
Walker2d	Limb length	$[0.95, 1.05]$	SAC

several different values of m_p and l, and collect all the resulting trajectories in a replay buffer $\mathcal{D}$. The expert here is the standard scenario-based MPC scheme of Schildbach et al. (2014), with a prediction horizon of $N = 20$ and 20 sampled scenarios. HeFLO is then trained entirely offline, using only the transitions stored in $\mathcal{D}$ and without further interaction with the underlying system. Since we minimize cost rather than maximize reward in this system, Figure 1 shows that the episodic cost falls sharply and converges over the course of HeFLO training. In Table 4, we also show the average normalized score (same as presented in Table 1) of HeFLO on the linearized InvertedPendulum environment.

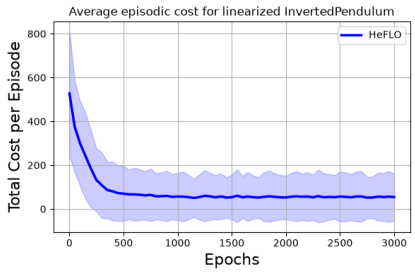


Figure 1: Total episodic cost vs epochs during HeFLO training.

Table 4: Average normalized score of HeFLO on Linearized InvertedPendulum

Environment	HeFLO
Linearized InvertedPendulum	99.41 ± 1.74

For this linear system we use linear models only. Table 5 gives the model architecture used for the linearized InvertedPendulum system.

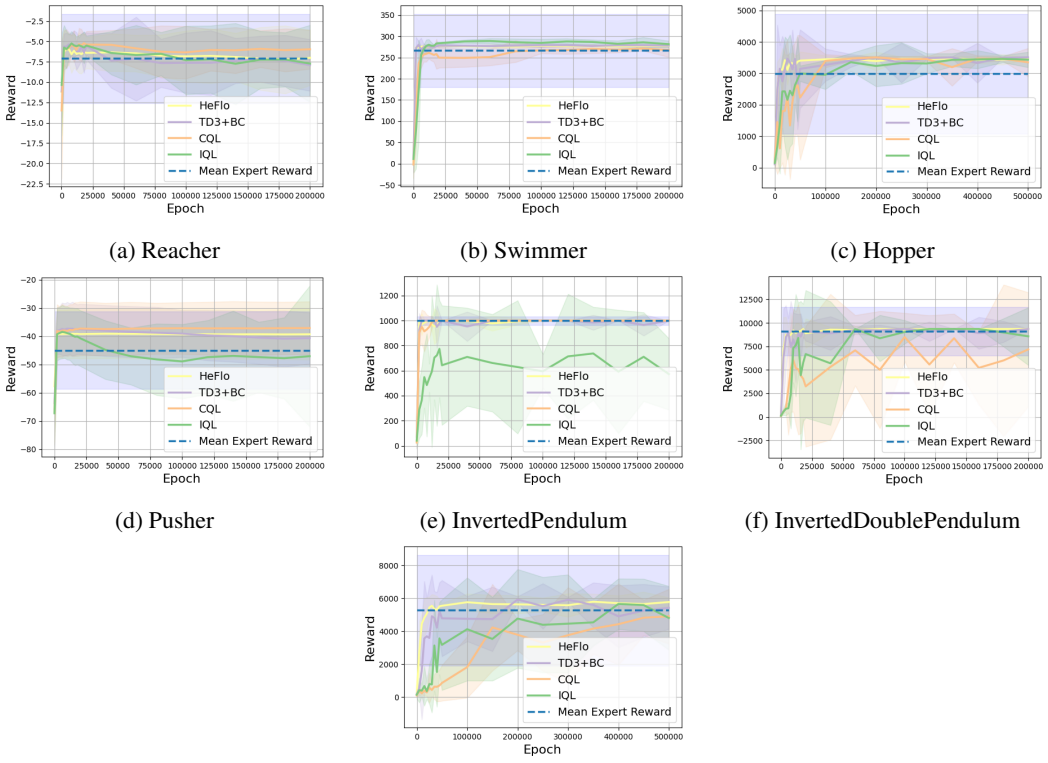
D.2 Convergence Analysis of HeFLO

Figure 2 shows the learning curves of HeFLO, CQL and TD3+BC during training; IQL is reported in Table 1 but omitted from the figure for legibility. To plot this figure, we run each algorithm for 20

Table 5: Model architectures and hyperparameters for Linearized InvertedPendulum

Model	Hyperparameters	Value
Flow model v_{θ_b}	Hidden layers	2
	Hidden dimension	$[256, 256]$
	Optimizer	SGD
	Learning rate	4×10^{-3}
Critic	Hidden layers	1
	Hidden dimension	256
	Optimizer	SGD
	Learning rate	4×10^{-3}
	Target update rate	5×10^{-3}
Actor	Hidden layers	1
	Hidden dimension	256
	Optimizer	SGD
	Learning rate	4×10^{-3}
BC Regularizer	α	0.1

episodes using 5 different seeds on each environment configuration, and average the rewards. Unlike Table 1, here we plot the raw reward from the environment instead of the normalized score.



(a) Reacher (b) Swimmer (c) Hopper

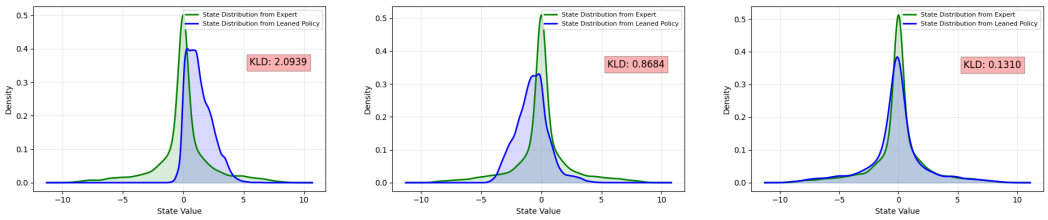
(d) Pusher (e) InvertedPendulum (f) InvertedDoublePendulum

(g) Walker2d

Figure 2: Learning curve of HeFLO compared to the other existing offline RL algorithms.

D.3 Effect of Behaviour Cloning Regularization

Assumption 1 posits that the Q-function approximation error remains bounded, provided the stationary state distribution of the learned policy closely matches that of the expert behaviour policy. To empirically validate this assumption, we plot the stationary distribution of the angular velocity of the first arm (state dimension 6) in the `Reacher-v5` environment at various stages of training in Figure 3. At the initial phase of training, the stationary distribution of the learned policy deviates significantly from that of the expert policy. However, as training progresses, the two distributions converge, with the learned policy's stationary distribution closely matching that of the expert by the later stages of training. This behaviour is consistent across all the state dimensions and environments we considered, providing empirical support for Assumption 1. To further quantify this convergence, we measure the Kullback-Leibler (KL) divergence between the two distributions directly from samples following (Pérez-Cruz, 2008), and observe a consistent decreasing trend in KL divergence over the course of training.

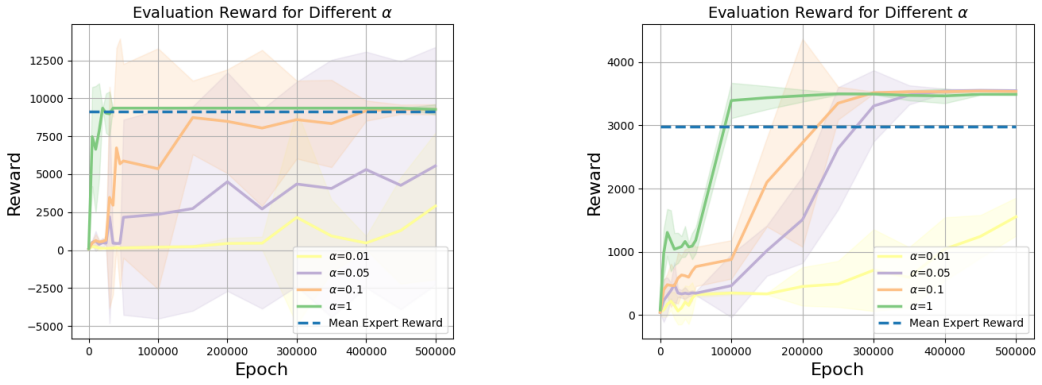


(a) training iteration: 10 (b) training iteration: 100 (c) training iteration: 50000

Figure 3: State distribution of the learned policy against that of the expert policy, at three points during training, for the angular velocity of the first arm (state dimension 6) in the `Reacher-v5` environment.

D.4 Effect of α on Training and Convergence

We investigate the effect of the regularisation parameter α on policy learning and convergence. Figure 4 illustrates the training behaviour for different values of α in two environments, `InvertedDoublePendulum` and `Hopper`. The results show that, when α exceeds an environment-dependent threshold, the policy consistently converges, although the number of epochs required for convergence varies with α. In contrast, when α falls below this threshold, the training process fails to converge. These empirical observations are consistent with Theorem 1, which establishes that policy convergence is guaranteed when α exceeds the threshold specified in (4.9).



(a) InverteDoublePendulum (b) Hopper

Figure 4: Learning curve of HeFLO compared to the other existing offline RL algorithms.

Table 6: Performance analysis of the two learned policies and the expert policy, with a 300-step episode limit.

Policy Details	Max step in Episode	Episodic Reward	# of Unhealthy States	Mean Velocity (m/s)	Mean Travel Distance (m)
Policy A	300	1564.34	12 (↑)	2.19	5.27
Policy B	300	1558.87	0	1.96 (↓)	4.73 (↓)
Expert Policy	300	1564.2	0	2.14	5.16

D.5 System-Level Analysis of Policies with Updated Cost Function

Table 6 presents a system-level performance comparison of three policies: (i) `Policy A`, trained on the default reward function of the environment as stored in the offline dataset $\mathcal{D}$; (ii) `Policy B`, trained on a modified reward function while using the same state-action pairs as in $\mathcal{D}$; and (iii) the `Expert Policy`, included as a reference baseline. The default reward function of the `Hopper-v5` environment is defined as (Brockman et al., 2016; Towers et al., 2024),

$$\begin{aligned} r &= r_{\text{healthy}} + r_{\text{forward}} - r_{\text{ctrl}} \\ &= r_{\text{healthy}} + w_{\text{forward}} \cdot \frac{dx}{dt} - w_{\text{ctrl}} \cdot \|u\|_2^2, \end{aligned}$$

where dx is the displacement of the torso, dt is the time between consecutive actions, and the default parameter values are $r_{\text{healthy}} = 1$, $w_{\text{forward}} = 1$, and $w_{\text{ctrl}} = 10^{-3}$. `Policy A` is trained on this default reward function, where the cost function is defined as its negation. For `Policy B`, we modify the reward function by setting $r_{\text{healthy}} = 5$ and $w_{\text{forward}} = 0.5$, and recompute the reward values directly from the trajectories stored in $\mathcal{D}$ without any further environment interaction. `Policy B` is then trained on this relabeled dataset, which shares the same state-action pairs as $\mathcal{D}$ but with updated reward values.

We evaluate all three policies over 20 independent trials, each consisting of 200 episodes of 300 steps and report the mean performance metrics in Table 6. We observe that `Policy A` enters an unhealthy state 12 times, whereas `Policy B` exhibits no such failures, which is attributable to the increased value of r_{healthy} during its training. Furthermore, the mean velocity and mean distance traveled by the hopper under `Policy B` are approximately 10% lower than those of `Policy A` and the `Expert Policy`, reflecting the effect of the reduced w_{forward}.

To further stress-test the learned policies, we extend the episode length to 500 steps and report the results in Table 7. Under this extended setting, `Policy A` enters the unhealthy state more frequently than `Policy B`, further corroborating the effect of the increased r_{healthy} term during the training of `Policy B`. Notably, the mean distance traveled by `Policy A` is lower than that of `Policy B` in this setting, which may appear counterintuitive. However, this is explained by the frequent early terminations of `Policy A` due to unhealthy states, which truncate episodes and reduce the total distance traveled. When comparing mean velocities directly, which are unaffected by early termination, `Policy B` exhibits an approximately 10% reduction relative to both `Policy A` and the `Expert Policy`, consistent with the effect of the reduced w_{forward} weight.

Table 7: Performance analysis of the two learned policies and the expert policy, with the episode limit extended to 500 steps.

Policy Details	Max step in Episode	Episodic Reward	# of Unhealthy Stage	Mean Velocity (m/s)	Mean Travel Distance (m)
Policy A	500	2293.56	105 (↑)	2.38	8.37
Policy B	500	2607.53	0	2.15 (↓)	8.63 (↓)
Expert Policy	500	2619.1	0	2.39	9.58

D.6 Experiments Compute Resource

All experiments were conducted on a personal laptop equipped with an Apple M4 Max chip and 64 GB of unified memory. The M4 Max features an integrated CPU–GPU architecture, and no external compute clusters or dedicated GPU servers were used. The implementation was executed in a standard local environment, and experiments were run sequentially without distributed training. The runtime varied across experiments depending on the specific task, model configuration, and hyperparameters, with runs consistently taking between one and three hours.

E Implementation Details

Our implementation of TD3+BC (Fujimoto & Gu, 2021) is based on the code released by the authors on GitHub[1]. Our implementations of CQL and IQL are based on the open-source library CORL (Tarasov et al., 2022) [2]. We use the same number of layers for every algorithm so that model complexity is comparable across methods; we note that the baselines were originally tuned for smaller networks, so this choice favours parity of capacity over per-baseline tuning. Table 8 gives the model architecture and hyperparameter values for HeFLO.

Table 8: Model architectures and hyperparameters

Model	Hyperparameters	Value
Flow model v_{θ_b}	Hidden layers	4
	Hidden dimension	512
	Activation	ReLU
	Optimizer	Adam
	Learning rate	3×10^{-4}
Critic	Hidden layers	4
	Hidden dimension	512
	Activation	ReLU
	Optimizer	Adam
	Learning rate	3×10^{-4}
	Target update rate	5×10^{-3}
Actor	Hidden layers	4
	Hidden dimension	512
	Activation	ReLU
	Optimizer	Adam
	Learning rate	3×10^{-4}
BC Regularizer	α	$[0.01, 1]$

Software. We use the following software to implement our algorithm:

- Python 3.13.2
- PyTorch 2.12.0 (Paszke et al., 2019)
- Gymnasium 1.2.3 (Towers et al., 2024)
- MuJoCo 3.9.0 (Todorov et al., 2012)

F Proof of Theoretical Results

This appendix contains the full statements and proofs supporting Theorem 1. Part (i) of the theorem is established here as the standalone Theorem 2, and the gradient-mismatch decomposition used in

[1] https://github.com/sfujim/TD3_BC
[2] https://github.com/tinkoff-ai/CORL

the main text is stated formally as Lemma 2. The organisation is as follows. Appendix F.1 fixes notation and records the two distinct objectives. Appendix F.2 establishes the quadratic structure of the value and action-value functions for the heterogeneous family (Proposition 1). Appendix F.3 proves the exact cost-difference identity (Lemma 1), from which the gradient formula (Lemma 2) and gradient dominance (Lemma 3) are both obtained without freezing the K-dependence of Θ_K. Appendix F.4 proves the stability result. Appendix F.5 proves smoothness, the finite-sample bound, and assembles the convergence theorem.

F.1 Setup and notation

Using equation 4.1 and $u_t = -Kx_t + \eta_t$ with $\eta_t \sim \mathcal{N}(0, \Sigma_\eta)$, $K \in \mathcal{K}$, the closed-loop dynamics of configuration δ are

$$x_{t+1} = A_K(\delta)x_t + h_t(\delta), \tag{F.1}$$

$$A_K(\delta) = A(\delta) - B(\delta)K, \quad h_t(\delta) = B(\delta)\eta_t + \omega_t,$$
$$\Sigma_h(\delta) = B(\delta)\Sigma_\eta B(\delta)^\top + \Sigma_\omega(\delta), \qquad \sigma_h := \inf_{\delta \in \mathcal{S}_\delta} \lambda_{\min}\big(\Sigma_h(\delta)\big) > 0, \tag{F.2}$$

where $\sigma_h > 0$ because $\Sigma_\omega(\delta) \succ 0$ is continuous on the compact set $\mathcal{S}_\delta$. For $K \in \mathcal{K}$ the closed loop is stable for every δ, so the stationary state covariance $\Sigma^K(\delta)$ of equation 4.2 exists, is unique, and satisfies

$$\Sigma^K(\delta) = \sum_{t \geq 0} A_K^t(\delta)\, \Sigma_h(\delta)\, \big(A_K^t(\delta)\big)^\top \;\succeq\; \Sigma_h(\delta) \;\succeq\; \sigma_h I. \tag{F.3}$$

The two objectives. We keep the following two quantities rigorously separate throughout. The *deployment cost* is

$$J(K) = \mathrm{Tr}\big[(R_x + K^\top R_u K)\,\bar{\Sigma}^K\big] + \mathrm{Tr}(R_u \Sigma_\eta), \qquad \bar{\Sigma}^K = \mathbb{E}_\delta\big[\Sigma^K(\delta)\big], \tag{F.4}$$

and the corresponding regularised deployment objective is

$$\mathcal{L}(K) = J(K) + \tfrac{\alpha}{2}\mathcal{R}(K), \text{where} \tag{F.5}$$

$$\mathcal{R}(K) = \mathbb{E}_{\delta \sim \rho_\delta,\, x \sim d_\mathcal{D},\, \eta \sim \mathcal{N}(0, \Sigma_\eta)} \big\| -Kx + \eta - \mu_{\theta_b}(x, \eta) \big\|_2^2. \tag{F.6}$$

Here $\Sigma_\mathcal{D} = \mathbb{E}_{x \sim d_\mathcal{D}}[xx^\top]$.

Conditioning on the realised dataset. Throughout the analysis, $d_\mathcal{D}$ is the *empirical* distribution of the $N_\mathcal{D}$ states in $\mathcal{D}$, each paired with a fixed noise draw η_i. All expectations over (x, η) in equation F.6 are therefore finite averages, and

$$\Sigma_\mathcal{D} = \frac{1}{N_\mathcal{D}} \sum_{i=1}^{N_\mathcal{D}} x_i x_i^\top, \qquad M_{\theta_b} = \frac{1}{N_\mathcal{D}} \sum_{i=1}^{N_\mathcal{D}} \mu_{\theta_b}(x_i, \eta_i)\, x_i^\top \tag{F.7}$$

are deterministic given $\mathcal{D}$: the flow model is frozen before policy learning, so the outputs $\mu_{\theta_b}(x_i, \eta_i)$ do not depend on K and are precomputed once. The regulariser gradient $\frac{\alpha}{2}\nabla_K \mathcal{R}(K) = \alpha\big(K\Sigma_\mathcal{D} + M_{\theta_b}\big)$ is consequently the same object in $\nabla_K \mathcal{L}$ and in the analysed update direction, and contributes no estimation error. The only randomness in the analysis is the critic estimate (Assumption 1) and the draw of the m configurations. Two consequences follow. First, the finite-sample error of the update direction (Lemma 5) carries no α-weighted terms, which is what makes ε_{tot} independent of α. Second, the dependence on $N_\mathcal{D}$ has not disappeared; it has moved into $\epsilon_\mathcal{D} = \|\bar{\Sigma}^{K_b} - \Sigma_\mathcal{D}\|_2$ of Corollary 1, which compares the empirical moment with a population covariance and is not multiplied by α. The analysed update uses the full-batch regulariser gradient; minibatching it is an unbiased stochastic approximation that the deterministic inexact-gradient argument below does not cover.

Note that $\mathcal{L}$ is the regularised deployment objective. As discussed in Section 3.2, the algorithm uses the update direction $g_\mathcal{L}$ in equation 3.6, which generally differs from the exact gradient of $\mathcal{L}$ or the offline surrogate. Our analysis bounds this discrepancy, $\|g_\mathcal{L}(K) - \nabla_K \mathcal{L}(K)\|_F$, to establish stability and convergence.

F.2 Structure of the value functions

We prove **Proposition 1**, which is stated in Section 4, as follows.

Proof. Fix δ. Under π_K the per-step expected cost at state x_t is

$$\mathbb{E}_{\eta_t}\big[c(x_t,u_t)\mid x_t\big] = x_t^\top (R_x + K^\top R_u K)x_t + \mathrm{Tr}(R_u\Sigma_\eta), \tag{F.8}$$

because the cross term $-2\,\mathbb{E}[\eta_t^\top R_u K x_t] = 0$ (η_t is zero-mean and independent of x_t). Unrolling equation F.1 from $x_0 = x$, $x_t = A_K^t(\delta)x + \sum_{i=0}^{t-1} A_K^{t-1-i}(\delta)h_i(\delta)$, and using independence of x and the h_i,

$$\mathbb{E}\big[c(x_t,u_t)\mid x_0 = x\big] = x^\top \big(A_K^t(\delta)\big)^\top (R_x + K^\top R_u K)\big(A_K^t(\delta)\big)x + c_t(\delta). \tag{F.9}$$

Here $c_t(\delta)$ does not depend on x and $\sum_{t\ge 0}\big(c_t(\delta) - J_\delta(K)\big)$ converges because $\rho(A_K(\delta)) < 1$. Substituting into $V_K^\delta(x) = \sum_{t\ge 0}\big\{\mathbb{E}[c(x_t,u_t)\mid x_0 = x] - J_\delta(K)\big\}$ yields equation 4.5, with constant term $v_c(\delta) = \sum_{t\ge 0}(c_t(\delta) - J_\delta(K))$; both series converge because $\rho(A_K(\delta)) < 1$. Separating the $t = 0$ term in the series for $P_K(\delta)$ gives its fixed-point equation.

For the action-value function, substitute $x_{t+1} = A(\delta)x + B(\delta)u + \omega$ into V_K^δ and take expectations over δ and ω:

$$\begin{aligned}
&\mathbb{E}_{\delta,\omega}\big[V_K^\delta(A(\delta)x + B(\delta)u + \omega)\big] \\
&= x^\top \mathbb{E}_\delta\big[A(\delta)^\top P_K(\delta)A(\delta)\big]x + 2x^\top \mathbb{E}_\delta\big[A(\delta)^\top P_K(\delta)B(\delta)\big]u \\
&\quad + u^\top \mathbb{E}_\delta\big[B(\delta)^\top P_K(\delta)B(\delta)\big]u + \mathbb{E}_{\delta,\omega}\big[\omega^\top P_K(\delta)\omega\big] + \mathbb{E}_\delta[v_c(\delta)],
\end{aligned} \tag{F.10}$$

where the ω cross terms vanish. Adding $c(x,u) - J(K)$ and collecting terms yields equation 4.6–equation 4.7 with $q_c = \mathbb{E}_{\delta,\omega}[\omega^\top P_K(\delta)\omega] + \mathbb{E}_\delta[v_c(\delta)] - J(K)$, which is finite because $P_K(\delta)$ is bounded uniformly on $\mathcal{S}_\delta$ for $K \in \mathcal{K}$.

Finally, $P_K(\delta)$ solves a discrete Lyapunov equation whose coefficients are polynomial in K, and the solution map of a Lyapunov equation is real-analytic wherever the closed loop is stable. The same argument holds for Σ^K. Uniformity in δ follows from continuity of A, B, Σ_ω on the compact set $\mathcal{S}_\delta$. ■

F.3 The cost-difference identity, the gradient, and gradient dominance

Lemma 1 (Cost difference). *Let $K, K' \in \mathcal{K}$ and write $\Delta := K' - K$. For each $\delta \in \mathcal{S}_\delta$,*

$$J_\delta(K') - J_\delta(K) = \mathrm{Tr}\Big[\Sigma^{K'}(\delta)\Big(\Delta^\top \Theta_K^{uu}(\delta)\,\Delta + 2\,\Delta^\top E_K(\delta)\Big)\Big], \tag{F.11}$$

where $\Theta_K^{uu}(\delta) := R_u + B(\delta)^\top P_K(\delta)B(\delta)$ and $E_K(\delta) := \Theta_K^{uu}(\delta)K - B(\delta)^\top P_K(\delta)A(\delta)$, and $J_\delta(K) := \mathrm{Tr}[(R_x + K^\top R_u K)\Sigma^K(\delta)] + \mathrm{Tr}(R_u\Sigma_\eta)$. Taking $\mathbb{E}_\delta$ gives the same identity for J.

Proof. We fix δ and drop it from the notation for brevity in the rest of the proof. Let $\{x_t\}$ be the stationary closed-loop process under K', so $x_t \sim \Sigma^{K'}$ and $x_{t+1} = A_{K'}x_t + h_t$. Using the Bellman equation for P_K, $P_K = R_x + K^\top R_u K + A_K^\top P_K A_K$, and the average-cost relation $J(K) = \mathrm{Tr}\big[(R_x + K^\top R_u K)\Sigma^K\big] + \mathrm{Tr}(R_u\Sigma_\eta) = \mathrm{Tr}\big[P_K\Sigma_h\big] + \mathrm{Tr}(R_u\Sigma_\eta)$, we compute, for any x,

$$\begin{aligned}
&x^\top\big(R_x + K'^\top R_u K'\big)x + \mathbb{E}_h\big[(A_{K'}x + h)^\top P_K(A_{K'}x + h)\big] - x^\top P_K x \\
&= x^\top\big(R_x + K'^\top R_u K' + A_{K'}^\top P_K A_{K'} - P_K\big)x + \mathrm{Tr}(P_K\Sigma_h).
\end{aligned}$$

Substituting $A_{K'} = A_K - B\Delta$ and $P_K = R_x + K^\top R_u K + A_K^\top P_K A_K$ and expanding, the quadratic form in parentheses equals

$$\begin{aligned}
&K'^\top R_u K' - K^\top R_u K + (A_K - B\Delta)^\top P_K(A_K - B\Delta) - A_K^\top P_K A_K \\
&= \Delta^\top(R_u + B^\top P_K B)\Delta + 2\Delta^\top\big[(R_u + B^\top P_K B)K - B^\top P_K A\big] \;=\; \Delta^\top\Theta_K^{uu}\Delta + 2\Delta^\top E_K,
\end{aligned}$$

where we used $K' = K + \Delta$ and $A_K = A - BK$. Now take expectations with $x \sim \Sigma^{K'}$ and average over t. The left-hand side telescopes: in stationarity $\mathbb{E}[x_{t+1}^\top P_K x_{t+1}] = \mathbb{E}[x_t^\top P_K x_t]$, so its time average is $J_\delta(K') - \mathrm{Tr}(R_u\Sigma_\eta) - \mathrm{Tr}(P_K\Sigma_h) = J_\delta(K') - J_\delta(K)$. Equating the two sides gives equation F.11. ■

Lemma 2 (Gradient mismatch). *Let $K \in \mathcal{K}$. Then the deployment cost equation 4.3 has gradient $\nabla_K J(K) = 2\,\mathbb{E}_\delta\big[E_K(\delta)\Sigma^K(\delta)\big]$, whereas the quantity computed by equation 3.5 is $g(K) = 2E_K\Sigma_{\mathcal{D}}$, with $E_K(\delta)$ and E_K as in equation 4.8. Consequently*

$$\nabla_K J(K) \;=\; g(K) \;+\; \Gamma_K, \qquad \Gamma_K := \underbrace{2E_K\big(\bar{\Sigma}^K - \Sigma_{\mathcal{D}}\big)}_{\textit{distribution shift}} + \underbrace{2\,\Xi_K}_{\textit{heterogeneity coupling}}, \tag{F.12}$$

where $\Xi_K := \mathbb{E}_\delta\big[(E_K(\delta) - E_K)(\Sigma^K(\delta) - \bar{\Sigma}^K)\big]$, and $\|\Gamma_K\|_F \le 2\|E_K\|_F\,\epsilon_{\mathrm{cov}} + 2\epsilon_{\mathrm{het}}$ with $\epsilon_{\mathrm{cov}} := \sup_{K\in\mathcal{B}}\|\bar{\Sigma}^K - \Sigma_{\mathcal{D}}\|_2$ and $\epsilon_{\mathrm{het}} := \sup_{K\in\mathcal{B}}\|\Xi_K\|_F$ over the fixed ball $\mathcal{B}$ of equation 4.4.

Proof of Lemma 2. *(a) The gradient of the deployment cost.* Put $K' = K + \varepsilon\Delta$ in equation F.11 for an arbitrary direction Δ, divide by ε and let $\varepsilon \downarrow 0$. The quadratic term is $O(\varepsilon^2)$ and $\Sigma^{K'}(\delta) \to \Sigma^K(\delta)$ by continuity of the Lyapunov solution (Proposition 1), so

$$\big\langle \nabla_K J_\delta(K), \Delta \big\rangle = 2\,\mathrm{Tr}\big[\Sigma^K(\delta)\,\Delta^\top E_K(\delta)\big] = \big\langle 2E_K(\delta)\Sigma^K(\delta),\, \Delta \big\rangle,$$

using symmetry of $\Sigma^K(\delta)$. As Δ was arbitrary, $\nabla_K J_\delta(K) = 2E_K(\delta)\Sigma^K(\delta)$ and $\nabla_K J(K) = 2\,\mathbb{E}_\delta[E_K(\delta)\Sigma^K(\delta)]$. We emphasise that this derivation is exact, as the K-dependence of $P_K(\delta)$ (and hence of Θ_K) is fully accounted for.

(b) The direction the algorithm computes. By equation 3.5 with the linear Gaussian policy $\pi_K(\cdot|x) = \mathcal{N}(-Kx, \Sigma_\eta)$, we have $\nabla_K \log \pi_K(u|x) = -\Sigma_\eta^{-1}(u + Kx)x^\top$, so

$$g(K) = -\,\mathbb{E}_{x\sim d_{\mathcal{D}},\,u\sim\pi_K}\big[Q_K(x,u)\,\Sigma_\eta^{-1}(u+Kx)x^\top\big] = -\,\mathbb{E}_{x\sim d_{\mathcal{D}},\,\eta}\big[\nabla_u Q_K(x,u)\,x^\top\big], \tag{F.13}$$

the second equality by Stein's lemma (Stein, 1981). From equation 4.6, $\nabla_u Q_K(x,u) = 2(\Theta_K^{ux}x + \Theta_K^{uu}u)$, and substituting $u = -Kx + \eta$ with $\mathbb{E}[\eta x^\top] = 0$ gives

$$g(K) = 2\big(\Theta_K^{uu}K - \Theta_K^{ux}\big)\Sigma_{\mathcal{D}} = 2E_K\Sigma_{\mathcal{D}}. \tag{F.14}$$

(c) The mismatch. Subtracting,

$$\begin{aligned}\nabla_K J(K) - g(K) &= 2\,\mathbb{E}_\delta\big[E_K(\delta)\Sigma^K(\delta)\big] - 2E_K\Sigma_{\mathcal{D}}\\ &= 2\,\mathbb{E}_\delta\big[(E_K(\delta) - E_K)(\Sigma^K(\delta) - \bar{\Sigma}^K)\big] + 2E_K\bar{\Sigma}^K - 2E_K\Sigma_{\mathcal{D}}\\ &= \;2\Xi_K + 2E_K(\bar{\Sigma}^K - \Sigma_{\mathcal{D}}),\end{aligned}$$

where we used $\mathbb{E}_\delta[E_K(\delta)] = E_K$ and $\mathbb{E}_\delta[\Sigma^K(\delta)] = \bar{\Sigma}^K$ so that the two cross terms cancel. The stated norm bound follows from $\|E_K(\bar{\Sigma}^K - \Sigma_{\mathcal{D}})\|_F \le \|E_K\|_F\|\bar{\Sigma}^K - \Sigma_{\mathcal{D}}\|_2$ and the triangle inequality. ∎

Lemma 3 (Gradient dominance). *Let $\mathcal{B}$ be the fixed ball of equation 4.4, with radius $r' < r_b$, so that $\mathcal{B}$ is convex and $\mathcal{B} \subset \mathcal{K}$, and let $L_J := \sup_{K\in\mathcal{B}}\|\nabla_K^2 J(K)\|_2 < \infty$ (finite by Proposition 1 and compactness of $\mathcal{B}$). Then*

(i) (Strongly regularized regime.) *If $\alpha > L_J/\sigma_{\min}$, then $\mathcal{L}$ is μ-strongly convex on $\mathcal{B}$ with $\mu := \alpha\sigma_{\min} - L_J > 0$, and consequently*

$$\mathcal{L}(K) - \mathcal{L}(K^\star) \;\le\; \frac{1}{2\mu}\big\|\nabla_K\mathcal{L}(K)\big\|_F^2 \qquad \forall\, K \in \mathcal{B}, \tag{F.15}$$

where $K^\star = \arg\min_{K\in\mathcal{B}}\mathcal{L}(K)$.

(ii) (Any α, LQR route.) *Without any condition on α, and for* any *comparator $K' \in \mathcal{K}$,*

$$J(K) - J(K') \;\le\; \frac{\bar{\sigma}^\star}{\lambda_{\min}(R_u)}\Big(\|E_K\|_F^2 + v_{\mathrm{het}}(K)\Big), \tag{F.16}$$

where $\bar{\sigma}^\star := \sup_\delta\|\Sigma^{K'}(\delta)\|_2$, and $v_{\mathrm{het}}(K) := \mathbb{E}_\delta\|E_K(\delta) - E_K\|_F^2$ is the heterogeneity variance of the policy-gradient matrix, which vanishes when $\mathcal{S}_\delta$ is a singleton.

Proof. *(i).* The regularizer equation F.6 is, as a function of K, an exact quadratic: writing $e(x,\eta) := \mu_{\theta_b}(x,\eta) - (-K_b x + \eta)$,

$$\tfrac{\alpha}{2}\mathcal{R}(K) = \tfrac{\alpha}{2}\Big(\mathrm{Tr}\big[(K-K_b)\Sigma_{\mathcal{D}}(K-K_b)^\top\big] + 2\,\mathrm{Tr}\big[(K-K_b)\,\mathbb{E}[xe^\top]\big] + \mathbb{E}\|e\|^2\Big), \tag{F.17}$$

(the sign of the cross term follows from $-Kx+\eta-\mu_{\theta_b}(x,\eta) = -\big((K-K_b)x + e(x,\eta)\big)$) whose Hessian acts as $\Delta \mapsto \alpha\,\Delta\,\Sigma_{\mathcal{D}}$, so that $\langle \nabla^2(\tfrac{\alpha}{2}\mathcal{R})[\Delta],\Delta\rangle = \alpha\,\mathrm{Tr}[\Delta\Sigma_{\mathcal{D}}\Delta^\top] \geq \alpha\sigma_{\min}\|\Delta\|_F^2$ by Assumption 2. Hence for every $K \in \mathcal{B}$ and every Δ,

$$\big\langle \nabla^2\mathcal{L}(K)[\Delta],\Delta\big\rangle \;\geq\; \alpha\sigma_{\min}\|\Delta\|_F^2 - L_J\|\Delta\|_F^2 \;=\; \mu\|\Delta\|_F^2.$$

Since $\mathcal{B}$ is convex, $\mathcal{L}$ is μ-strongly convex on $\mathcal{B}$, and equation F.15 is the standard Polyak–Łojasiewicz consequence of strong convexity.

(ii). Apply Lemma 1 with the comparator $K' \in \mathcal{K}$ and $\Delta = K' - K$. For each δ, completing the square,

$$\Delta^\top\Theta_K^{uu}(\delta)\Delta + 2\Delta^\top E_K(\delta) = \big(\Delta + \Theta_K^{uu}(\delta)^{-1}E_K(\delta)\big)^\top\Theta_K^{uu}(\delta)\big(\cdot\big) - E_K(\delta)^\top\Theta_K^{uu}(\delta)^{-1}E_K(\delta),$$

and since $\Sigma^{K'}(\delta) \succeq 0$ and $\Theta_K^{uu}(\delta) \succ 0$, the first term contributes non-negatively to the trace. Therefore

$$J(K') - J(K) \;\geq\; -\,\mathbb{E}_\delta\mathrm{Tr}\big[\Sigma^{K'}(\delta)\,E_K(\delta)^\top\Theta_K^{uu}(\delta)^{-1}E_K(\delta)\big] \;\geq\; -\,\frac{\bar\sigma^\star}{\lambda_{\min}(R_u)}\,\mathbb{E}_\delta\|E_K(\delta)\|_F^2,$$

using $\Theta_K^{uu}(\delta) \succeq R_u$. Finally the bias–variance split $\mathbb{E}_\delta\|E_K(\delta)\|_F^2 = \|E_K\|_F^2 + v_{\mathrm{het}}(K)$ gives equation F.16. ■

F.4 Stability of the iterates

The following result is part (i) of Theorem 1.

Theorem 2 (Iterate confinement)**.** *Let Assumptions 2 and 3 hold, let $K_0 \in \mathcal{K}$, and let $r_b := \sup\{r > 0 : \mathbb{B}(K_b, r) \subset \mathcal{K}\}$, which is strictly positive. Fix r' and $\mathcal{B}$ as in equation 4.4, put $D' := (\sqrt{\sigma_{\min}}r' - \epsilon_b)^2$ and $\Delta_{\mathcal{L}} := 4\varepsilon_{\mathrm{tot}}^2/\mu$, and let $\gamma \leq 1/(2L)$. Suppose the flow model is accurate enough, the initialization good enough, and the regularizer strong enough that*

$$\epsilon_b < \sqrt{\sigma_{\min}}\,r', \qquad \mathcal{R}(K_0) < D', \qquad \alpha \;>\; \max\Big\{\frac{2\big(J(K_0)+\Delta_{\mathcal{L}}\big)}{D'-\mathcal{R}(K_0)},\; \frac{2L_J}{\sigma_{\min}}\Big\}. \tag{F.18}$$

Then (a) every K with $\mathcal{L}(K) \leq \mathcal{L}(K_0) + \Delta_{\mathcal{L}}$ satisfies $\|K - K_b\|_F \leq r(\alpha) \leq r' < r_b$, with $r(\alpha) := \big(\epsilon_b + \sqrt{2(J(K_0)+\Delta_{\mathcal{L}})/\alpha + \mathcal{R}(K_0)}\big)/\sqrt{\sigma_{\min}}$, and hence $K \in \mathcal{K}$; and (b) the iterates $K_{k+1} = K_k - \gamma\hat g_{\mathcal{L}}(K_k)$ of Algorithm 1 satisfy $\mathcal{L}(K_k) \leq \mathcal{L}(K_0) + \Delta_{\mathcal{L}}$ for every $k \geq 0$ on the event of Lemma 5. Consequently $K_k \in \mathcal{B} \subset \mathcal{K}$ for all k, i.e. the closed loop is stable for every $\delta \in \mathcal{S}_\delta$ *at every intermediate iterate.*

Proof of Theorem 2. *Step 1: $r_b > 0$.* The map $(K,\delta) \mapsto \rho\big(A(\delta) - B(\delta)K\big)$ is continuous, and $\mathcal{S}_\delta$ is compact, so $K \mapsto \sup_{\delta\in\mathcal{S}_\delta}\rho(A_K(\delta))$ is continuous. Since $K_b \in \mathcal{K}$ we have $\sup_\delta \rho(A_{K_b}(\delta)) =: 1-\theta$ with $\theta > 0$, and by continuity there is $r_b > 0$ such that $\sup_\delta \rho(A_K(\delta)) < 1$ for all $\|K - K_b\|_F \leq r_b$. Hence $\mathbb{B}(K_b, r_b) \subset \mathcal{K}$ and the supremum defining r_b is over a non-empty set.

Step 2: a sublevel bound implies a norm bound. Let K satisfy $\mathcal{L}(K) \leq c + \Delta_{\mathcal{L}}$ with $c := \mathcal{L}(K_0) = J(K_0) + \frac{\alpha}{2}\mathcal{R}(K_0)$. Since $J \geq 0$,

$$\tfrac{\alpha}{2}\,\mathcal{R}(K) \;\leq\; \mathcal{L}(K) \;\leq\; c + \Delta_{\mathcal{L}}, \qquad \text{so} \qquad \mathcal{R}(K) \;\leq\; \frac{2\big(J(K_0)+\Delta_{\mathcal{L}}\big)}{\alpha} + \mathcal{R}(K_0). \tag{F.19}$$

Write $e(x,\eta) := \mu_{\theta_b}(x,\eta) - (-K_b x + \eta)$, so that by Assumption 3 $\|e\|_{L^2} \leq \epsilon_b$, where $\|\cdot\|_{L^2}$ denotes the norm of $L^2(d_{\mathcal{D}} \otimes \mathcal{N}(0,\Sigma_\eta))$. Since $-Kx + \eta - \mu_{\theta_b}(x,\eta) = (K_b - K)x - e(x,\eta)$, the reverse triangle inequality in L^2 gives

$$\sqrt{\mathcal{R}(K)} \;=\; \big\|(K_b - K)x - e\big\|_{L^2} \;\geq\; \big\|(K-K_b)x\big\|_{L^2} - \epsilon_b. \tag{F.20}$$

By Assumption 2,

$$\big\|(K-K_b)x\big\|_{L^2}^2 = \mathrm{Tr}\big[(K-K_b)\Sigma_{\mathcal{D}}(K-K_b)^\top\big] \;\geq\; \sigma_{\min}\|K-K_b\|_F^2. \tag{F.21}$$

Combining equation F.19–equation F.21, $\sqrt{\sigma_{\min}}\,\|K-K_b\|_F - \epsilon_b \leq \sqrt{\mathcal{R}(K)}$, i.e.

$$\|K-K_b\|_F \;\leq\; \frac{\epsilon_b + \sqrt{2(J(K_0)+\Delta_{\mathcal{L}})/\alpha + \mathcal{R}(K_0)}}{\sqrt{\sigma_{\min}}} \;=:\; r(\alpha). \tag{F.22}$$

Step 3: the threshold. The third condition of equation F.18 is equivalent to $2(J(K_0)+\Delta_{\mathcal{L}})/\alpha < D' - \mathcal{R}(K_0)$, i.e. to $2(J(K_0)+\Delta_{\mathcal{L}})/\alpha + \mathcal{R}(K_0) < D' = (\sqrt{\sigma_{\min}}r' - \epsilon_b)^2$, which by equation F.22 gives $r(\alpha) < r'$. The requirement $\epsilon_b < \sqrt{\sigma_{\min}}r'$ is exactly what makes D' positive, and the requirement $\mathcal{R}(K_0) < D'$ is exactly what makes the third condition satisfiable by some finite α. If instead $\mathcal{R}(K_0) \geq D'$, no α works, however large. Note also that $\varepsilon_{\mathrm{tot}}$ does not depend on α. By Lemma 5 the constants c_1, c_4 are α-free, and the remaining terms $2G\epsilon_{\mathrm{cov}}+2\epsilon_{\mathrm{het}}$ are suprema over the fixed ball $\mathcal{B}$ of equation 4.4. Since in addition $\mu \geq \alpha\sigma_{\min}/2$ (using $\alpha > 2L_J/\sigma_{\min}$), we get $\Delta_{\mathcal{L}} \leq 8\varepsilon_{\mathrm{tot}}^2/(\alpha\sigma_{\min})$, so the left-hand side of the third condition is $O(\alpha^{-1})$. Therefore the condition is satisfiable for all sufficiently large α whenever $\mathcal{R}(K_0) < D'$. Hence $K \in \mathbb{B}(K_b, r(\alpha)) \subset \mathbb{B}(K_b, r') = \mathcal{B} \subset \mathcal{K}$.

Step 4: the iterates, by induction. We show $\mathcal{L}(K_k) \leq c + \Delta_{\mathcal{L}}$ for all k, where now $c := \mathcal{L}(K_0)$ and $\Delta_{\mathcal{L}} = 4\varepsilon_{\mathrm{tot}}^2/\mu$. The base case $k=0$ is immediate. Assume $\mathcal{L}(K_k) \leq c + \Delta_{\mathcal{L}}$. Steps 2–3, applied at level $c + \Delta_{\mathcal{L}}$, then give $K_k \in \mathcal{B}$, so Lemmas 4 and 3(i) are available at K_k and the one-step inequality equation F.30 holds, with $\nabla_k := \nabla_K \mathcal{L}(K_k)$:

$$\mathcal{L}(K_{k+1}) \;\leq\; \mathcal{L}(K_k) - \tfrac{\gamma}{4}\|\nabla_k\|_F^2 + \tfrac{3\gamma}{2}\varepsilon_{\mathrm{tot}}^2.$$

We distinguish two regimes. *(a) Outside the floor:* if $\|\nabla_k\|_F^2 \geq 6\varepsilon_{\mathrm{tot}}^2$ then $\frac{\gamma}{4}\|\nabla_k\|_F^2 \geq \frac{3\gamma}{2}\varepsilon_{\mathrm{tot}}^2$, so $\mathcal{L}(K_{k+1}) \leq \mathcal{L}(K_k) \leq c + \Delta_{\mathcal{L}}$ and the induction closes. *(b) Inside the floor:* if $\|\nabla_k\|_F^2 < 6\varepsilon_{\mathrm{tot}}^2$, then gradient dominance equation F.15 gives $\mathcal{L}(K_k) - \mathcal{L}(K^\star) \leq \frac{1}{2\mu}\|\nabla_k\|_F^2 < 3\varepsilon_{\mathrm{tot}}^2/\mu$, and $\mathcal{L}(K^\star) = \min_{\mathcal{B}} \mathcal{L} \leq \mathcal{L}(K_0) = c$ because $K_0 \in \mathcal{B}$. Hence $\mathcal{L}(K_k) < c + 3\varepsilon_{\mathrm{tot}}^2/\mu$ and, discarding the negative term in the one-step inequality,

$$\mathcal{L}(K_{k+1}) \;<\; c + \frac{3\varepsilon_{\mathrm{tot}}^2}{\mu} + \frac{3\gamma}{2}\varepsilon_{\mathrm{tot}}^2 \;\leq\; c + \frac{3\varepsilon_{\mathrm{tot}}^2}{\mu} + \frac{3\varepsilon_{\mathrm{tot}}^2}{4\mu} \;<\; c + \frac{4\varepsilon_{\mathrm{tot}}^2}{\mu} \;=\; c + \Delta_{\mathcal{L}},$$

where we used $\gamma \leq 1/(2L) \leq 1/(2\mu)$. The induction closes in both regimes, so $\mathcal{L}(K_k) \leq c + \Delta_{\mathcal{L}}$ and therefore $K_k \in \mathcal{B} \subset \mathcal{K}$ for every k, i.e. $\rho(A_{K_k}(\delta)) < 1$ for every $\delta \in \mathcal{S}_\delta$ simultaneously. Note that the argument uses the smoothness and gradient-dominance estimates only at iterates already known to lie in $\mathcal{B}$, which is what removes the circularity. ■

Remark 5. The slack $\Delta_{\mathcal{L}}$ is proportional to the squared error floor and hence vanishes in the exact-gradient limit, in which case the statement reduces to the familiar sublevel-set argument. Note also that no claim of monotone decrease of $\mathcal{L}$ is made or needed because inside the floor $\mathcal{L}$ may oscillate, and the induction only rules out escape from the enlarged sublevel set.

Corollary 1 (The regularizer controls the distribution shift)**.** *Let L_Σ be the Lipschitz constant of $K \mapsto \bar{\Sigma}^K$ on $\mathcal{B}$* (finite by Proposition 1) *and let $\epsilon_{\mathcal{D}} := \|\bar{\Sigma}^{K_b} - \Sigma_{\mathcal{D}}\|_2$ be the expert-aggregation gap. Then, under the conditions of Theorem 2,*

$$\sup_k \big\|\bar{\Sigma}^{K_k} - \Sigma_{\mathcal{D}}\big\|_2 \;\leq\; L_\Sigma\, r(\alpha) \;+\; \epsilon_{\mathcal{D}} \qquad \textit{and} \qquad \epsilon_{\mathrm{cov}} \;\leq\; L_\Sigma\, r' + \epsilon_{\mathcal{D}}, \tag{F.23}$$

the second bound following from $r(\alpha) \leq r'$. If in addition the actor is warm-started by behaviour cloning, so that $\mathcal{R}(K_0) \leq \epsilon_b^2$ (Remark 1), then $r(\alpha) \leq \big(2\epsilon_b + \sqrt{2(J(K_0)+\Delta_{\mathcal{L}})/\alpha}\big)/\sqrt{\sigma_{\min}}$ and

$$\sup_k \big\|\bar{\Sigma}^{K_k} - \Sigma_{\mathcal{D}}\big\|_2 \;=\; O\big(\epsilon_b + \alpha^{-1/2}\big) \;+\; \epsilon_{\mathcal{D}}. \tag{F.24}$$

Proof. By the triangle inequality, $\|\bar{\Sigma}^{K_k} - \Sigma_{\mathcal{D}}\|_2 \leq \|\bar{\Sigma}^{K_k} - \bar{\Sigma}^{K_b}\|_2 + \|\bar{\Sigma}^{K_b} - \Sigma_{\mathcal{D}}\|_2 \leq L_\Sigma\|K_k - K_b\|_F + \epsilon_{\mathcal{D}}$, and $\|K_k - K_b\|_F \leq r(\alpha)$ by equation F.22; the bound on ϵ_{cov} is the same computation with $r(\alpha)$ replaced by the radius r' of $\mathcal{B}$. The warm-started form follows from $\sqrt{a+\epsilon_b^2} \leq \sqrt{a} + \epsilon_b$. ■

Remark 6. Based on Corollary 1, increasing α (i) enforces stability through equation 4.9, (ii) shrinks the distribution-shift contribution $2G\epsilon_{\text{cov}}$ to the error floor equation 4.11 at rate $\alpha^{-1/2}$, and (iii) enlarges the strong-convexity modulus $\mu = \alpha\sigma_{\min} - L_J$ of Lemma 3(i), which shrinks the floor $\propto 1/\mu$. Note that increasing α does not always speed up convergence without qualification. The admissible step size $\gamma \le 1/(2L)$ with $L = L_J + \alpha\sigma_{\max}$ shrinks like α^{-1}, and as computed in the step-size discussion at the end of Appendix F.5 the per-step contraction $\beta = 1 - \gamma\mu/2$ saturates at $1 - \sigma_{\min}/(4\sigma_{\max})$. Beyond that point additional regularisation buys stability margin and a smaller floor, but no additional speed. Against this, a larger α moves the minimiser $K^\star$ of $\mathcal{L}$ away from the minimiser of J and towards K_b. Proposition 2 bounds that displacement and shows the resulting deployment-cost bias stays bounded as $\alpha \to \infty$, so the trade-off is not as sharp as it appears. Note also the residuals ϵ_b and $\epsilon_{\mathcal{D}}$ in equation F.23, which no amount of regularisation removes.

F.5 Smoothness, finite-sample error, and proof of Theorem 1

Lemma 4 (Smoothness). *On the fixed convex compact set $\mathcal{B} = \mathbb{B}(K_b, r') \subset \mathcal{K}$ of equation 4.4, $\mathcal{L}$ is L-smooth with*

$$L \;=\; L_J + \alpha\,\sigma_{\max}, \qquad L_J := \sup_{K\in\mathcal{B}} \big\|\nabla_K^2 J(K)\big\|_2 < \infty. \tag{F.25}$$

Proof. By Proposition 1, J is real-analytic on the open set $\mathcal{K}$, hence twice continuously differentiable there; $\mathcal{B}$ is a compact subset of $\mathcal{K}$, so by the Weierstrass theorem $L_J = \sup_{\mathcal{B}} \|\nabla^2 J\|_2$ is finite. The regularizer is the exact quadratic equation F.17 with Hessian $\Delta \mapsto \alpha\Delta\Sigma_{\mathcal{D}}$, whose operator norm is $\alpha\|\Sigma_{\mathcal{D}}\|_2 \le \alpha\sigma_{\max}$. Adding the two and applying the mean value inequality along the segment joining any $K, K' \in \mathcal{B}$ (which lies in $\mathcal{B}$ by convexity) gives $\|\nabla\mathcal{L}(K) - \nabla\mathcal{L}(K')\|_F \le L\|K - K'\|_F$. ■

Lemma 5 (Finite-sample error of the update direction). *Let the idealized update direction be $g_{\mathcal{L}}(K) := 2E_K\Sigma_{\mathcal{D}} + \alpha K\Sigma_{\mathcal{D}} + \alpha M_{\theta_b}$ with $\Sigma_{\mathcal{D}}$ and M_{θ_b} the empirical moments equation F.7, and let the computed direction be $\hat{g}_{\mathcal{L}}(K) := 2(\hat{\Theta}_K^{uu}K - \hat{\Theta}_K^{ux})\Sigma_{\mathcal{D}} + \alpha K\Sigma_{\mathcal{D}} + \alpha M_{\theta_b}$, which differs from $g_{\mathcal{L}}$ only through the critic estimate of Assumption 1, the m configurations $\delta_1, \dots, \delta_m \overset{\text{iid}}{\sim} \rho_\delta$ present in $\mathcal{D}$ being the only other source of randomness. Then under Assumptions 1 and 2, simultaneously for all $K \in \mathcal{B}$,*

$$\sup_{K\in\mathcal{B}} \big\|\hat{g}_{\mathcal{L}}(K) - g_{\mathcal{L}}(K)\big\|_F \;\le\; c_1\epsilon_Q + c_4\epsilon_m \qquad \text{w.p. } 1 - p_K, \tag{F.26}$$

where $p_K := p_Q + p_m$ and

$$c_1 = 2\big(1 + \sup_{\mathcal{B}}\|K\|_2\big)\|\Sigma_{\mathcal{D}}\|_2, \qquad c_4 = 2\|\Sigma_{\mathcal{D}}\|_2, \qquad \epsilon_m = O\Big(\sqrt{\tfrac{n_u n_x + \log(1/p_m)}{m}}\Big).$$

In particular c_1 and c_4 depend only on $\sup_{\mathcal{B}}\|K\|_2$ and $\|\Sigma_{\mathcal{D}}\|_2$, and not *on α; and $\epsilon_m \to 0$ as $m \to \infty$ but is not reduced by additional transitions from the same experts.*

Proof. Write $E_K = \Theta_K^{uu}K - \Theta_K^{ux}$ and $\hat{E}_K = \hat{\Theta}_K^{uu}K - \hat{\Theta}_K^{ux}$. Since the empirical moments $\Sigma_{\mathcal{D}}$ and M_{θ_b} of equation F.7 appear identically in $\hat{g}_{\mathcal{L}}$ and $g_{\mathcal{L}}$, the two regularizer terms cancel and

$$\hat{g}_{\mathcal{L}}(K) - g_{\mathcal{L}}(K) = 2\big(\hat{E}_K - E_K\big)\Sigma_{\mathcal{D}}. \tag{F.27}$$

It remains to bound $\|\hat{E}_K - E_K\|_F$ uniformly on $\mathcal{B}$, which we do by splitting it into the critic error relative to the empirical mixture and the generalization from that mixture to ρ_δ.

The finite-m term. The critic is trained on transitions pooled over the m configurations present in $\mathcal{D}$, so what it can identify is the empirical mixture $\Theta_K^{(m)}$ obtained by replacing $\mathbb{E}_{\delta\sim\rho_\delta}$ in equation 4.7 by $\frac{1}{m}\sum_{j=1}^m$. Since $(K,\delta) \mapsto B(\delta)^\top P_K(\delta)A(\delta)$ and $(K,\delta) \mapsto B(\delta)^\top P_K(\delta)B(\delta)$ are continuous on the compact set $\mathcal{B} \times \mathcal{S}_\delta$ (Proposition 1), they are bounded there by some C_E and Lipschitz in K with some constant L_E. Hoeffding's inequality for bounded independent matrices, applied on an ε-net of $\mathcal{B}$ of cardinality $(3r'/\varepsilon)^{n_u n_x}$ and combined with the Lipschitz property to pass from the net to all of $\mathcal{B}$, gives

$$\sup_{K\in\mathcal{B}} \big\|E_K^{(m)} - E_K\big\|_F \;\le\; C\,C_E\sqrt{\frac{n_u n_x \log(3r'L_E m) + \log(1/p_m)}{m}} \;=:\; \epsilon_m \tag{F.28}$$

with probability at least $1 - p_m$, where C is a universal constant. This is the only term in the bound that is driven by the number of demonstrated configurations rather than by the number of transitions, and it does not decrease when more data are collected from the same experts.

Uniformity in K. The same ε-net argument upgrades the preceding display from a pointwise to a uniform statement over $\mathcal{B}$, since $K \mapsto \hat{g}_{\mathcal{L}}(K) - g_{\mathcal{L}}(K)$ is Lipschitz on $\mathcal{B}$ with a constant polynomial in $\sup_{\mathcal{B}} \|K\|_2$ and $\|\Sigma_{\mathcal{D}}\|_2$. Uniformity is what allows the bound to be invoked at the data-dependent iterates K_k, and is why Assumption 1 is stated uniformly.

The critic term. By the triangle inequality, $\|\hat{E}_K - E_K\|_F \le \|\hat{E}_K - E_K^{(m)}\|_F + \|E_K^{(m)} - E_K\|_F$, where $E_K^{(m)} := \Theta_K^{(m),uu} K - \Theta_K^{(m),ux}$. For the first summand, $\|\hat{E}_K - E_K^{(m)}\|_F \le \|\hat{\Theta}_K - \Theta_K^{(m)}\|_F(1 + \|K\|_2) \le \epsilon_Q(1 + \|K\|_2)$ uniformly on $\mathcal{B}$ with probability $1 - p_Q$ by Assumption 1, whose target is exactly $\Theta_K^{(m)}$. The second summand is at most ϵ_m by equation F.28, with probability $1 - p_m$; note that ϵ_m is defined directly on E, so it carries no factor $(1 + \|K\|_2)$, and that it enters the bound exactly once. On the intersection of the two events, whose probability is at least $1 - (p_Q + p_m)$ by a union bound, equation F.27 and $\|MN\|_F \le \|M\|_F \|N\|_2$ give equation F.26 with $c_1 = 2(1 + \sup_{\mathcal{B}} \|K\|_2)\|\Sigma_{\mathcal{D}}\|_2$ and $c_4 = 2\|\Sigma_{\mathcal{D}}\|_2$. Finally, ϵ_m can be made small only by increasing the number of demonstrated configurations m, whereas ϵ_Q absorbs all transition-level finite-sample effects through the critic. ■

Proof of Theorem 1. Part (i) is Theorem 2(b), whose hypotheses are exactly equation 4.9 and whose proof establishes the required sublevel bound by induction along the inexact iterates (Step 4 of its proof), so no monotonicity of $\mathcal{L}$ is assumed. Note that the induction and the present argument use the same event, namely the one of Lemma 5, so no additional union bound over iterations is incurred and the conclusion holds for all k simultaneously with probability at least $1 - p_K$. It remains to prove part (ii). By Theorem 2 all iterates lie in the convex compact set $\mathcal{B} = \mathbb{B}(K_b, r') \subset \mathcal{K}$, on which Lemma 4 gives L-smoothness and Lemma 3(i) gives μ-gradient dominance for $\mu = \alpha\sigma_{\min} - L_J > 0$.

Step 1: total gradient error. The algorithm takes $K_{k+1} = K_k - \gamma_k \hat{g}_{\mathcal{L}}(K_k)$. Decompose the deviation from the true gradient of the deployment objective as

$$\hat{g}_{\mathcal{L}}(K_k) - \nabla_K \mathcal{L}(K_k) = \underbrace{\big[\hat{g}_{\mathcal{L}}(K_k) - g_{\mathcal{L}}(K_k)\big]}_{\text{finite sample, Lemma 5}} - \underbrace{\Gamma_{K_k}}_{\text{mismatch, Lemma 2}}, \tag{F.29}$$

where we used $\nabla_K \mathcal{L}(K) = \nabla_K J(K) + \frac{\alpha}{2}\nabla_K \mathcal{R}(K) = g(K) + \Gamma_K + \alpha K \Sigma_{\mathcal{D}} + \alpha M_{\theta_b} = g_{\mathcal{L}}(K) + \Gamma_K$. Note that the regularizer contributes identically to $g_{\mathcal{L}}$ and to $\nabla\mathcal{L}$ and therefore cancels from the error, so the entire mismatch is attributable to the J term. By Lemmas 2 and 5 and the triangle inequality, $\sup_{K \in \mathcal{B}} \|\hat{g}_{\mathcal{L}}(K) - \nabla_K \mathcal{L}(K)\|_F \le \varepsilon_{\text{tot}}$ on a single event of probability at least $1 - p_K$, where ε_{tot} is as in equation 4.11 and $G = \sup_{K \in \mathcal{B}} \|E_K\|_F < \infty$ by compactness. The uniform bound applies to all data-dependent iterates K_k simultaneously, avoiding a per-iteration union bound whose guarantee $1 - kp_K$ becomes uninformative for the iteration counts used in Section 5.

Step 2: one step of inexact gradient descent. Write $\nabla_k := \nabla_K \mathcal{L}(K_k)$ and $\hat{g}_k := \hat{g}_{\mathcal{L}}(K_k) = \nabla_k + e_k$ with $\|e_k\|_F \le \varepsilon_{\text{tot}}$. By L-smoothness,

$$\begin{aligned}\mathcal{L}(K_{k+1}) &\le \mathcal{L}(K_k) - \gamma_k \langle \nabla_k, \hat{g}_k \rangle + \tfrac{L\gamma_k^2}{2}\|\hat{g}_k\|_F^2 \\ &\le \mathcal{L}(K_k) - \gamma_k \|\nabla_k\|_F^2 + \gamma_k \|\nabla_k\|_F \varepsilon_{\text{tot}} + L\gamma_k^2\big(\|\nabla_k\|_F^2 + \varepsilon_{\text{tot}}^2\big),\end{aligned}$$

using $\|a + b\|^2 \le 2\|a\|^2 + 2\|b\|^2$. For $\gamma_k \le \frac{1}{2L}$ we have $L\gamma_k^2 \|\nabla_k\|_F^2 \le \frac{\gamma_k}{2}\|\nabla_k\|_F^2$, and by Young's inequality $\gamma_k \|\nabla_k\|_F \varepsilon_{\text{tot}} \le \frac{\gamma_k}{4}\|\nabla_k\|_F^2 + \gamma_k \varepsilon_{\text{tot}}^2$. Hence, using $L\gamma_k \le 1/2$,

$$\mathcal{L}(K_{k+1}) \;\le\; \mathcal{L}(K_k) - \tfrac{\gamma_k}{4}\|\nabla_k\|_F^2 + \tfrac{3\gamma_k}{2}\varepsilon_{\text{tot}}^2. \tag{F.30}$$

Step 3: apply gradient dominance. Substituting $\|\nabla_k\|_F^2 \ge 2\mu\big(\mathcal{L}(K_k) - \mathcal{L}(K^\star)\big)$ from equation F.15 into equation F.30 and subtracting $\mathcal{L}(K^\star)$,

$$\mathcal{L}(K_{k+1}) - \mathcal{L}(K^\star) \;\le\; \Big(1 - \tfrac{\gamma_k \mu}{2}\Big)\big(\mathcal{L}(K_k) - \mathcal{L}(K^\star)\big) + \tfrac{3\gamma_k}{2}\varepsilon_{\text{tot}}^2 \quad \text{on the event of Lemma 5,} \tag{F.31}$$

which is the claimed one-step contraction with $\beta = 1 - \gamma_k \mu/2 \in (0, 1)$ (note $\mu \le L$ so $\gamma_k \mu/2 \le 1/4$). Iterating equation F.31 with a constant step size $\gamma_k \equiv \gamma$ and summing the geometric series gives

$$\mathcal{L}(K_k) - \mathcal{L}(K^\star) \;\le\; \beta^k\big(\mathcal{L}(K_0) - \mathcal{L}(K^\star)\big) + \frac{3\gamma\varepsilon_{\text{tot}}^2/2}{1 - \beta} \;=\; \beta^k\big(\mathcal{L}(K_0) - \mathcal{L}(K^\star)\big) + \frac{3\varepsilon_{\text{tot}}^2}{\mu}, \tag{F.32}$$

simultaneously for all k on the event of Lemma 5, i.e. with probability at least $1 - p_K$, which is equation 4.10. This proves part (ii), and hence Theorem 1. ■

F.6 Proof of Proposition 2

Write $f := \mathcal{L}$, which by Lemma 3(i) is μ-strongly convex on the convex set $\mathcal{B}$ with constrained minimizer $K^\star$. For any $K \in \mathcal{B}$, first-order optimality gives $\langle \nabla f(K^\star), K - K^\star \rangle \geq 0$, and strong convexity gives $\langle \nabla f(K) - \nabla f(K^\star), K - K^\star \rangle \geq \mu \|K - K^\star\|_F^2$. Adding and applying the Cauchy–Schwarz inequality,

$$\|K - K^\star\|_F \;\leq\; \frac{1}{\mu} \big\|\nabla f(K)\big\|_F, \qquad \forall K \in \mathcal{B}. \tag{F.33}$$

Apply equation F.33 at $K = K_J^\star$, which lies in $\mathcal{B}$ by hypothesis. Since $K_J^\star$ minimizes J over the open set $\mathcal{K}$ we have $\nabla_K J(K_J^\star) = 0$, so by equation F.17,

$$\nabla f(K_J^\star) \;=\; \tfrac{\alpha}{2} \nabla_K \mathcal{R}(K_J^\star) \;=\; \alpha \Big[(K_J^\star - K_b)\Sigma_{\mathcal{D}} + \mathbb{E}\big[e(x,\eta)\, x^\top\big] \Big].$$

By Assumption 2, $\|(K_J^\star - K_b)\Sigma_{\mathcal{D}}\|_F \leq \sigma_{\max} \|K_J^\star - K_b\|_F$, and by the Cauchy–Schwarz inequality together with Assumption 3,

$$\big\|\mathbb{E}[e\, x^\top]\big\|_F \;\leq\; \sqrt{\mathbb{E}\|e\|^2}\, \sqrt{\mathbb{E}\|x\|^2} \;\leq\; \epsilon_b \sqrt{\operatorname{Tr} \Sigma_{\mathcal{D}}} \;=\; \epsilon_b c_x.$$

Substituting into equation F.33 gives the first inequality of equation 4.12. For the second, J is L_J-smooth on the convex set $\mathcal{B}$ (Lemma 4) and $\nabla_K J(K_J^\star) = 0$, so $J(K^\star) - J(K_J^\star) \leq \frac{L_J}{2} \|K^\star - K_J^\star\|_F^2$. Finally $\alpha/\mu = \alpha/(\alpha\sigma_{\min} - L_J) \downarrow 1/\sigma_{\min}$ as $\alpha \to \infty$, which gives the stated limit. ■

Discussion of Proposition 2. The bound is uniform in α, and this is the point. One might expect the regularisation bias to grow without limit as α increases, since $K^\star \to K_b$. What equation 4.12 shows, however, is that the limit of that displacement is $\|K^\star - K_J^\star\|_F \leq \frac{\sigma_{\max}}{\sigma_{\min}} \|K_J^\star - K_b\|_F + \frac{c_x}{\sigma_{\min}} \epsilon_b$, i.e. the displacement is bounded by the suboptimality of the demonstrated behaviour, amplified by the condition number $\sigma_{\max}/\sigma_{\min}$ of the dataset covariance. Two consequences are worth recording. First, when the demonstrations are themselves optimal for the family, the price of arbitrarily strong regularisation is zero, so the stability certificate of Theorem 1(i) may be bought at no cost in deployment performance. Second, when they are not, the same condition number that governs the convergence rate also governs the bias, which identifies dataset conditioning as the single quantity controlling both.

F.7 Discussion on the step size

Lemma 4 gives $L = L_J + \alpha\sigma_{\max}$, so the admissible step size $\gamma \leq 1/(2L)$ shrinks as the regularisation α grows, while the contraction factor $\beta = 1 - \gamma\mu/2$ improves as $\mu = \alpha\sigma_{\min} - L_J$ grows. At the optimal choice $\gamma = 1/(2L)$ the per-step contraction is $\beta = 1 - \frac{\alpha\sigma_{\min} - L_J}{4(L_J + \alpha\sigma_{\max})}$, which tends to $1 - \sigma_{\min}/(4\sigma_{\max})$ as $\alpha \to \infty$. In other words, the rate saturates at a value governed by the conditioning of the dataset covariance, $\sigma_{\min}/\sigma_{\max}$. This is an interpretable statement about offline learning: how fast one can learn is limited by how evenly the demonstrations excite the state space. Furthermore, it is the heterogeneous, regularized analogue of the dependence on the conditioning of Σ^K in the on-policy LQR analysis of Fazel et al. (2018). □

F.8 Discussion on the effect of the heterogeneity range $\mathcal{S}_\delta$.

We note that the set $\mathcal{S}_\delta$ does not appear explicitly in the contraction factor $\beta = 1 - \gamma\mu/2$. In other words, heterogeneity does not change the geometric nature of the convergence, only the constants and the floor. It affects the convergence analysis in the following ways.

(i) *Feasibility.* $\mathcal{K}$ is an intersection over $\delta \in \mathcal{S}_\delta$, so it shrinks as $\mathcal{S}_\delta$ widens, and so does r_b. The threshold equation 4.9 grows like $(\sqrt{\sigma_{\min}} r_b - \epsilon_b)^{-2}$ and diverges as $r_b \downarrow \epsilon_b/\sqrt{\sigma_{\min}}$; below that point no single linear gain uniformly stabilizes the family, and Assumption 3 becomes unsatisfiable. Since the demonstrations come from per-configuration experts, a wider $\mathcal{S}_\delta$ also increases ϵ_b, tightening $\epsilon_b < \sqrt{\sigma_{\min}} r_b$ from both sides. Moreover L_J, hence $\mu = \alpha\sigma_{\min} - L_J$, degrades with the width.

(ii) *Error floor.* Both factors of Ξ_K are deviations from their δ-means and $\delta \mapsto E_K(\delta), \Sigma^K(\delta)$ are Lipschitz on the compact $\mathcal{S}_\delta$ (Proposition 1), so $\epsilon_{\text{het}} = O(\text{diam}(\mathcal{S}_\delta)^2)$ and the floor $3\varepsilon_{\text{tot}}^2/\mu$ degrades as $\text{diam}(\mathcal{S}_\delta)^4$, vanishing exactly when $\mathcal{S}_\delta$ is a singleton.

(iii) *Irreducible suboptimality.* In equation F.16 the variance $v_{\text{het}}(K)$ does not vanish at a stationary point of the surrogate, where $E_K = 0$. It is the price of committing to one gain for a family whose per-configuration optima disagree, and it is not removed by more data, a better critic, or a larger α. Together with $\epsilon_{\mathcal{D}}$ in equation F.23 and ϵ_m in equation F.28, it is one of three quantities in this analysis that are properties of the demonstration set rather than of the algorithm.

(iv) *Two regimes.* Items (i) and (ii) act in opposite directions on the practitioner's choice: widening $\mathcal{S}_\delta$ improves coverage of the deployment distribution and reduces ϵ_m at fixed m only if the additional configurations are genuinely new, while it simultaneously shrinks r_b, raises the α threshold, and inflates ϵ_{het} and $\epsilon_{\mathcal{D}}$. The analysis therefore predicts an interior optimum in the width of $\mathcal{S}_\delta$ for a fixed budget of experts, which we leave to future work.

□