2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Data-Efficient Hierarchical Goal-Conditioned Reinforcement Learning via Normalizing Flows

Shaswat Garg[1], Matin Moezzi[1] and Brandon Da Silva[1]

Abstract— **Hierarchical goal-conditioned reinforcement learning (H-GCRL) provides a powerful framework for tackling complex, long-horizon tasks by decomposing them into structured subgoals. However, its practical adoption is hindered by poor data efficiency and limited policy expressivity, especially in offline or data-scarce regimes. In this work, Normalizing flow-based hierarchical implicit Q-learning (NF-HIQL), a novel framework that replaces unimodal gaussian policies with expressive normalizing flow policies at both the high- and low-levels of the hierarchy is introduced. This design enables tractable log-likelihood computation, efficient sampling, and the ability to model rich multimodal behaviors. New theoretical guarantees are derived, including explicit KL-divergence bounds for Real-valued non-volume preserving (RealNVP) policies and PAC-style sample efficiency results, showing that NF-HIQL preserves stability while improving generalization. Empirically, NF-HIQL is evaluted across diverse long-horizon tasks in locomotion, ball-dribbling, and multi-step manipulation from OGBench. NF-HIQL consistently outperforms prior goal-conditioned and hierarchical baselines, demonstrating superior robustness under limited data and highlighting the potential of flow-based architectures for scalable, data-efficient hierarchical reinforcement learning.**

I. INTRODUCTION

Generalization is a cornerstone of Reinforcement learning (RL): it empowers agents to tackle new, unseen tasks and adapt to ever-changing environments [1]. Despite remarkable progress, deep RL agents now master continuous control challenges like locomotion [2], dexterous object manipulation [3], and robotic arm coordination [4], these successes have mostly focused on short-horizon, low-level motor skills executed in isolation. Real-world problems, by contrast, demand hierarchical reasoning: agents must integrate perception, planning, and decision-making across multiple layers of abstraction and compose primitive behaviors into long-horizon strategies. H-GCRL provides a natural framework for tackling complex, long-horizon tasks by decomposing them into sequences of subgoals, each managed by a corresponding low-level policy. This hierarchical structure allows agents to reason over multiple timescales and compose simpler behaviors into more sophisticated strategies. Recent advances in offline Goal-conditioned RL (GCRL) [5] and hierarchical extensions [6] demonstrate that it is possible to train such agents using large, unlabeled datasets such as videos or multi-task demonstrations by conditioning policies on desired goals, even when explicit reward or action information is missing.

However, collecting such diverse datasets is often infeasible due to high acquisition costs and safety concerns. This makes it essential to develop algorithms that generalize effectively from limited data, leveraging structure and inductive biases inherent in the task space.

To tackle the challenge of sample inefficiency [7], recent approaches have primarily focused on leveraging powerful generative models, which also provide the side benefit of greater policy expressiveness. Diffusion models offer rich expressiveness and have shown promise in capturing complex action distributions, but they are computationally expensive due to the need to solve differential equations during training and inference [8], [9]. Autoregressive models scale more efficiently and allow parallel training, yet they often rely on learning discrete representations of actions, which can introduce quantization artifacts and complicate optimization [10]. In contrast, gaussian policies are lightweight and efficient to train but lack the capacity to represent multimodal or structured behaviors, limiting their effectiveness in hierarchical or goal-conditioned settings [11].

In this work, a novel approach is proposed that leverages Normalizing flows (NFs), specifically the RealNVP architecture within the Hierarchical implicit Q-learning (HIQL) framework to bridge this gap [12]. NFs strike a favorable balance between expressivity and computational tractability [13]. In particular, they provide exact likelihood evaluation, which yields unbiased and lower-variance gradient estimates during training, leading to more stable optimization and improved sample efficiency compared to methods that rely on approximate likelihoods such as MCMC-based or variational approaches (e.g., energy-based models), a property that is especially advantageous in offline and data-constrained regimes [14]. A natural question is why NFs are preferred over diffusion models. In hierarchical offline RL, diffusion policies face two key disadvantages: (i) multi-step iterative denoising at inference, adding latency that compounds at both hierarchical levels, and (ii) lack of exact log-likelihoods, requiring variational bounds that introduce bias into advantage-weighted regression. NFs offer single-pass sampling with exact density evaluation, enabling unbiased gradient computation and efficient inference at every hierarchical decision step. By integrating normalizing flows into both the high-level and low-level policies of a hierarchical framework, a more expressive and data-efficient alternative is introduced to traditional policy representations. Furthermore, a theoretical analysis is presented that the learned policies are bounded in KL divergence and enjoy provable guarantees on sample efficiency, pushing the boundaries of scalable and robust long-horizon decision-making.

Shaswat Garg, Matin Moezzi and Brandon Da Silva are with ArenaX Labs. GitHub repository: https://github.com/Shaswat2001/heirarchical_RL

II. RELATED WORK

This section provides an overview of recent advancements in HGCRL and GCRL, with a particular focus on efforts to improve sample efficiency, especially through the use of generative models.

GCRL enables agents to learn a spectrum of tasks by conditioning on a goal input, fostering generalization across different outcomes [5]. A major challenge, however, is sample inefficiency in sparse-reward settings. Relabeling strategies like Hindsight experience replay (HER) [15] and density-based goal sampling [16] address this by reusing or prioritizing goals. Generative approaches also improve efficiency, e.g., learning latent dynamics models for planning [17], synthesizing goal-directed rollouts with GANs [18], or incorporating planning into offline GCRL [19]. While these methods generate additional data or plans, they rely on accurate learned models or GAN training, which can be brittle and hard to scale. In contrast, the proposed work sidesteps explicit planning by leveraging expressive policy models and hierarchical value-based learning to improve efficiency directly.

Long-horizon goal-reaching tasks benefit from hierarchical decomposition, where high-level subgoals improve learning efficiency [20], [21]. HIQL [12] extends this by learning a single goal-conditioned value function offline and deriving both high- and low-level policies, with subgoals proposed in latent space. This provides clearer learning signals and outperforms prior offline GCRL methods. Theory also supports hierarchy as a way to reduce sample complexity [22]. Yet, most HGCRL methods, including HIQL rely on simple Gaussian policies, limiting their ability to capture complex, multimodal behaviors, and neglect modern generative models. The proposed method addresses this by using normalizing flows, enabling richer hierarchical policies while retaining tractable training and efficient sampling.

Recent advances in generative modeling for RL have introduced more expressive policy classes to improve sample efficiency. Diffusion models, for instance, have been applied to goal-conditioned settings: [23] proposed a diffusion-based policy that achieves strong offline performance via denoising-based inference. Generative Flow Networks (GFlowNets) offer trajectory-level diversity, as in Goal2FlowNets [24], which enhance generalization. However, both diffusion models and GFlowNets involve complex training and sampling, leading to high computational costs. Normalizing flows (NFs) provide a tractable alternative [25]; SAC-NF [26] demonstrated improved convergence and expressivity by replacing Gaussian policies with NFs.

However, existing applications of NFs have primarily focused on flat policy architectures. The proposed work bridges this gap by integrating NFs into both the high-level and low-level policies of the HIQL framework. This yields expressive, multimodal policies at each level of hierarchy while preserving the tractable training and efficient sampling that NFs offer. In doing so, the method introduced enhances both the generalization capacity and sample efficiency of HGCRL in complex, long-horizon environments. Compared to goal-conditioned imitation learning methods such as BESO [27], which treat states as goals and learn policies conditioned directly on state-space targets using diffusion models, our approach leverages expressive normalizing flows to model the distribution over goals. In addition to the inherent advantages of normalizing flows over diffusion models, this formulation enables a broader abstraction level for the goal representation beyond raw state-space targets.

III. BACKGROUND

The problem is framed as a Markov decision process (MDP) [28] and a dataset $\mathcal{D}$, defined by a tuple $< S, A, \mu, P, R, \gamma >$, where S represents the set of possible states, A represents the set of possible actions, $\mu \in \mathcal{P}(\mathcal{S})$ denotes an initial state distribution, $P : S \times A \to S$ is the state transition function that represents the conditional probability $P(\mathbf{s}'|\mathbf{s}, \mathbf{a})$ or deterministic function $\mathbf{s}' = P(\mathbf{s}, \mathbf{a})$, and $R : S \times G \to \mathbb{R}$ represents the goal-conditioned reward function. The dataset $\mathcal{D}$ consists of trajectories $\tau = (s_0, a_0, s_1, a_1, \ldots, s_T)$. It is assumed that the goal space $\mathcal{G}$ is identical to the state space, that is, $\mathcal{G} = \mathcal{S}$. The aim is to learn a goal-conditioned policy $\pi(a \mid s, g)$ using $\mathcal{D}$ such that the expected cumulative reward

$$J(\pi) = \mathbb{E}_{g\sim p(g),\, \tau\sim p^{\pi}(\tau)} \left[\sum_{t=0}^{T} \gamma^t R(s_t, g) \right] \tag{1}$$

is maximized. The trajectory distribution under policy π is given by

$$p^{\pi}(\tau) = \mu(s_0) \prod_{t=0}^{T-1} \pi(a_t \mid s_t, g)\, P(s_{t+1} \mid s_t, a_t), \tag{2}$$

where γ denotes the discount factor and $p(g)$ is the distribution over goals.

IV. ALGORITHM

We consider a HGCRL setup following HIQL [12], where a single value function $V(s, g)$ guides two policies: a high-level subgoal policy π^h and a low-level action policy π^ℓ. The high-level policy proposes a future state (latent subgoal) s_{t+k} given the current state s_t and goal g, while the low-level policy selects actions a_t to reach it. In NF-HIQL, both policies are modeled as conditional normalizing flows, enabling exact density, gradient, and entropy computation while supporting expressive, multimodal behaviors.

A. Flow-Based Policy Parameterization

Concretely as shown in Algorithm 1, each policy is defined by an invertible function that maps a noise vector to an output (subgoal or action). For example, the high-level policy is given as follows:

$$s_{t+k} = f_H(u;\, s_t, g), \quad u \sim \mathcal{N}(0, I), \tag{3}$$

where $f_H(\cdot; s_t, g)$ is a neural-network flow conditioned on (s_t, g). In other words, latent noise u is sampled from

a standard Gaussian and passed through f_H to produce a candidate subgoal s_{t+k}. Similarly, the low-level policy uses its own flow f_ℓ to map $v \sim \mathcal{N}(0, I)$ (conditioned on the current state and chosen subgoal) to an action: $a_t = f_\ell(v; s_t, s_{t+k})$.

Because f_H and f_ℓ are bijective with tractable Jacobians, the log-density of an output can be computed exactly via the change-of-variables formula. For instance, if $u = f_H^{-1}(s_{t+k}; s_t, g)$, then

$$\log \pi^h(s_{t+k} \mid s_t, g) = \log p_H(u) - \log\Big|\det(\frac{\partial f_H(u;\, s_t, g)}{\partial u})\Big|. \quad (4)$$

Here $p_H(u)$ is the Gaussian base density (e.g. $\mathcal{N}(0, I)$), and $\det(\partial f_H/\partial u)$ is the Jacobian determinant of the flow. An analogous formula holds for the low-level policy: if $v = f_\ell^{-1}(a_t; , s_t, s_{t+k})$, then

$$\log \pi^\ell(a_t \mid s_t, s_{t+k}) = \log p_\ell(v) - \log\Big|\det(\frac{\partial f_\ell(v;\, s_t, s_{t+k})}{\partial v})\Big|. \quad (5)$$

In short, flow transforms endow the policy with an analytic log-probability while providing high expressivity. By stacking invertible layers, a simple base density is transformed into a richer, potentially multimodal distribution. In this work, RealNVP [14] serves as a universal approximator for continuous densities, enabling policies that capture complex multimodal action or subgoal distributions while supporting efficient sampling and exact likelihood evaluation.

B. Log-Probability, Entropy, and Advantage-Weighted Objectives

Because the flow policies admit exact densities, the usual advantage-weighted learning objectives can be written in the closed form. As in HIQL, the high-level advantage is defined as $A^h(s_t, s_{t+k}, g) = V(s_{t+k}, g) - V(s_t, g)$ and the low-level advantage as $A^\ell(s_t, a_t, s_{t+1}, s_{t+k}) = V(s_{t+1}, s_{t+k}) - V(s_t, s_{t+k})$. Then the weighted maximum-likelihood (AWR-style) objectives are:

- High-level objective:

$$J^h(\theta_H) = \mathbb{E}_{\text{data}}\Big[e^{\beta A^h}\, \log \pi^h_{\theta_H}(s_{t+k} \mid s_t, g)\Big]. \quad (6)$$

- Low-level objective:

$$J^\ell(\theta_L) = \mathbb{E}_{\text{data}}\Big[e^{\beta A^\ell}\, \log \pi^\ell_{\theta_L}(a_t \mid s_t, s_{t+k})\Big]. \quad (7)$$

Each expectation is over logged offline transitions (with subgoals s_{t+k} and actions a_t) with weight $\exp(\beta A)$, so that higher-advantage samples are upweighted. Substituting the flow log-densities above makes both J^h and J^ℓ fully differentiable functions of the flow parameters θ_H, θ_L. In particular, the gradients take the simple form as follows:

$$\begin{aligned} \nabla_{\theta_H} J^h &= \mathbb{E}[e^{\beta A^h}\, \nabla_{\theta_H} \log \pi^h(s_{t+k} \mid s_t, g)], \\ \nabla_{\theta_L} J^\ell &= \mathbb{E}[e^{\beta A^\ell}\, \nabla_{\theta_L} \log \pi^\ell(a_t \mid s_t, s_{t+k})]. \end{aligned} \quad (8)$$

Since $\log \pi^h$ and $\log \pi^\ell$ are given in closed form by the flow (the only trainable part of $\log p(u)$ is constant), then $\nabla_\theta \log \pi = -\nabla_\theta[\log|\det(\partial f/\partial u)|]$. Thus no policy sampling or likelihood-ratio estimators are needed: the Jacobian log-det gradient can be computed analytically for each data point.

Because the flows yield exact densities, the policy entropies can also be computed in closed form if desired. For example,

$$\begin{aligned} H(\pi^h) &= -\mathbb{E}_{s\sim\pi^h}[\log \pi^h(s)] \\ &= -\mathbb{E}_{u\sim p_H}[\log p_H(u) - \log|\det(\partial f_H/\partial u)|], \end{aligned} \quad (9)$$

which can be estimated by sampling $u \sim p_H$ (all terms inside are known). In short, all key quantities, log-likelihoods and entropies are tractable and exact for the flow policies.

Algorithm 1 Offline HIQL with Normalizing Flow Policies

Require: Dataset $\mathcal{D}$, Networks: value function $V_{\theta_V}(s, g)$, target value function $V_{\bar{\theta}_V}(s, g)$, high-level policy $\pi^h_{\theta_H}(s_{t+k}|s_t, g)$, low-level policy $\pi^\ell_{\theta_L}(a_t|s_t, s_{t+k})$, Hyper parameters: learning rates α_V, α_H, α_L; temperature β

while not converged **do**
  // 1. Update Value Function using action-free IQL
  Sample (s_t, s_{t+1}, g) from $\mathcal{D}$
  $y \leftarrow r(s_t, g) + \gamma V_{\bar{\theta}_V}(s_{t+1}, g)$
  $\theta_V \leftarrow \theta_V - \alpha_V \nabla_{\theta_V} \rho_\tau \left(y - V_{\theta_V}(s_t, g)\right)$
  // 2. Update High-Level Policy (Normalizing Flow)
  Sample (s_t, s_{t+k}, g) from $\mathcal{D}$
  $A^h \leftarrow V_{\theta_V}(s_{t+k}, g) - V_{\theta_V}(s_t, g)$
  Compute $\log \pi^h_{\theta_H}(s_{t+k} \mid s_t, g)$ via flow:
    $u = f_H^{-1}(s_{t+k}; s_t, g)$
    $\log \pi^h = \log p_H(u) - \log|\det \partial f_H/\partial u|$
  $\theta_H \leftarrow \theta_H + \alpha_H \nabla_{\theta_H} \left[e^{\beta A^h} \log \pi^h\right]$
  // 3. Update Low-Level Policy (Normalizing Flow)
  Sample $(s_t, a_t, s_{t+1}, s_{t+k})$ from $\mathcal{D}$
  $A^\ell \leftarrow V_{\theta_V}(s_{t+1}, s_{t+k}) - V_{\theta_V}(s_t, s_{t+k})$
  Compute $\log \pi^\ell_{\theta_L}(a_t \mid s_t, s_{t+k})$ via flow:
    $v = f_\ell^{-1}(a_t; s_t, s_{t+k})$
    $\log \pi^\ell = \log p_\ell(v) - \log|\det \partial f_\ell/\partial v|$
  $\theta_L \leftarrow \theta_L + \alpha_L \nabla_{\theta_L} \left[e^{\beta A^\ell} \log \pi^\ell\right]$
end while

C. Theoretical Guarantees

To complement the algorithmic design, NF-HIQL is supported by theoretical results on stability and efficiency. First, an upper bound on the KL divergence between the hierarchical policy and the behavior policy in $\mathcal{D}$ ensures the learned policy stays close to the data distribution, mitigating out-of-distribution actions and extrapolation error. Second, a PAC-style sample efficiency bound shows that the hierarchical policies converge toward the advantage-weighted target distribution used during training, with an explicit rate that depends on the dataset size and policy class capacity. We emphasize that these bounds characterize proximity to the advantage-weighted target rather than to the globally optimal policy; the gap to the true optimum additionally depends on

(a) (b) (c) (d) (e)

Fig. 1: Evaluation environments: (a) AntMaze—medium-navigate (long-horizon maze navigation); (b) AntSoccer—medium-navigate (wall-bounded dribbling and navigation); (c) AntSoccer—arena-navigate (open-field dribbling and navigation); (d) Cube—single-play (pick-and-place from play data); (e) Scene—play (multi-object, multi-step sequencing from play) [29].

the offline data coverage and the value function approximation error ε_V. Together, these results establish NF-HIQL as both stable and provably sample-efficient, reinforcing its practicality for real-world settings with limited data. Full technical proofs are provided in Appendix VII-A.

Lemma 2 (KL Divergence Bound)**.** *Let $\pi^b(\cdot \mid s)$ denote the behavior policy and $\pi_\theta(\cdot \mid s)$ the learned RealNVP policy given state s. If the action space is bounded and the behavior density is capped by a constant $M < \infty$, then there exists a constant $B < \infty$ (determined by the RealNVP architecture) such that*

$$\mathrm{KL}(\pi^b(\cdot \mid s) \,\|\, \pi_\theta(\cdot \mid s)) \;\le\; B + \log M. \tag{10}$$

Proof. See Appendix VII-A. □

Lemma 3 (PAC-Style Sample Efficiency)**.** *Let $\hat{\pi}_h, \hat{\pi}_\ell$ be the policies learned by advantage-weighted Maximum Likelihood Estimation (MLE). With probability at least $1-\delta$,*

$$\begin{aligned} &J(\pi^\star) - J(\hat{\pi}_{h,\ell}) \\ &\le \frac{H_h A_{\max,h}}{1-\gamma} \sqrt{\frac{C_h}{2}\left(4\,\mathfrak{R}_{n_h}(\mathcal{F}_h) + 2B_h\sqrt{\frac{\log(2/\delta)}{2n_h}}\right)} \\ &\quad + \frac{H_\ell A_{\max,\ell}}{1-\gamma} \sqrt{\frac{C_\ell}{2}\left(4\,\mathfrak{R}_{n_\ell}(\mathcal{F}_\ell) + 2B_\ell\sqrt{\frac{\log(2/\delta)}{2n_\ell}}\right)} \\ &\quad + \varepsilon_V, \end{aligned} \tag{11}$$

Proof. See Appendix VII-B. □

V. RESULTS

NF-HIIQL is evaluated on five OGBench tasks spanning long-horizon locomotion, ball-dribbling, and multi-step manipulation: antmaze-medium-navigate, antsoccer-medium-navigate, antsoccer-arena-navigate, cube-single-play, and scene-play (Figure 1). We follow OGBench's official environment definitions, dataset splits, and multi-goal success-rate protocol for offline goal-conditioned RL [29]. Each algorithm is trained on the same offline dataset for 1M transitions with five random seeds on NVIDIA T4 GPUs. Unless noted otherwise, we use the state-based benchmark variants, adapted to the benchmark's evaluation goals (five per manipulation environment) when reporting success rates and confidence intervals.

Comparisons include three representative offline GCRL baselines from OGBench, GCIQL, CRL, and HIQL along with the diffusion-based BESO [27], providing a strong reference set across navigation and manipulation. Two additional flow-based variants are considered: NF-HIQL (hierarchical flow policies at both levels) and NF-GCIQL (a flow policy under the GCIQL objective). Consistent with the motivation for expressive, multimodal policies, NF-HIQL outperforms all baselines.

To assess sample efficiency, each experiment is conducted in two regimes: (i) training on 100% of the available dataset and (ii) training on exactly 50% of the same dataset (a uniform halving of trajectories). These paired settings let us quantify the sample-efficiency gap between flow-based methods and unimodal counterparts.

NF-HIQL's success rates under both regimes are reported in Table I and Table II. With 100% data, NF-HIQL matches or exceeds all baselines: it achieves $95\pm2\%$ on `antmaze-medium-navigate` (vs. HIQL $96\pm1\%$, BESO $85\pm7\%$), and $73\pm1\%$ on `antsoccer-arena-navigate` (vs. HIQL $58\pm2\%$, BESO $56\pm2\%$), a 26% relative gain over the strongest baseline.

The key distinction emerges at 50% data, where competing methods degrade sharply while NF-HIQL remains robust. On `antsoccer-arena-navigate`, NF-HIQL maintains $73\pm4\%$ while HIQL collapses to $1\pm1\%$ and BESO drops to $30\pm2\%$. In `scene-play`, NF-HIQL achieves $36\pm3\%$ ($6\times$ HIQL, $2.5\times$ BESO). Across all tasks, NF-HIQL trained on 50% data often matches or exceeds baselines trained on the full dataset, with gains most pronounced in multimodal tasks. Detailed results are provided in the supplementary video.

Table III shows that using flows at both levels consistently yields the best performance. The high-level-only variant outperforms low-level-only on navigation (multimodal subgoal distributions matter for planning), while low-level-only is competitive on manipulation (expressive action distributions aid fine motor control). Both levels together capture complementary benefits.

Table IV shows performance peaks at $L{=}4$ coupling layers, with diminishing returns beyond $L{=}6$ due to increased optimization difficulty.

NF-HIQL adds $\sim$18% training time over HIQL (4.0h vs. 3.4h for 1M transitions on NVIDIA T4), but inference is $\sim5\times$ faster than BESO (single-pass vs. 10–50 denoising steps). BESO training takes 6.2h.

Environment	BESO	GCIQL	CRL	HIQL	NF-GCIQL	NF-HIQL
antmaze-medium-navigate	85 ±7	71 ±4	95 ±1	**96** ±1	82 ±3	95 ±2
antsoccer-medium-navigate	12 ±3	7 ±1	3 ±1	13 ±1	6 ±4	**14** ±2
antsoccer-arena-navigate	56 ±2	50 ±2	23 ±2	58 ±2	30 ±3	**73** ±1
cube-single-play	21 ±2	68 ±6	19 ±2	15 ±3	**70** ±1	37 ±2
scene-play	81 ±3	51 ±4	19 ±2	38 ±3	50 ±2	**40** ±3

TABLE I: Overall success rate (%) across all the tasks — dataset size: **100%** of the available offline dataset.

Environment	BESO	GCIQL	CRL	HIQL	NF-GCIQL	NF-HIQL
antmaze-medium-navigate	63 ±6	24 ±2	50 ±2	58 ±4	64 ±3	**72** ±4
antsoccer-medium-navigate	1 ±1	0 ±0	0 ±0	0 ±0	**7** ±2	3 ±2
antsoccer-arena-navigate	30 ±2	2 ±1	0 ±0	1 ±1	41 ±4	**73** ±4
cube-single-play	4 ±1	10 ±6	6 ±3	4 ±2	**40** ±9	36 ±4
scene-play	14 ±2	8 ±3	2 ±2	6 ±4	33 ±4	**36** ±3

TABLE II: Overall success rate (%) across all the tasks — dataset size: **50%** of the available offline dataset.

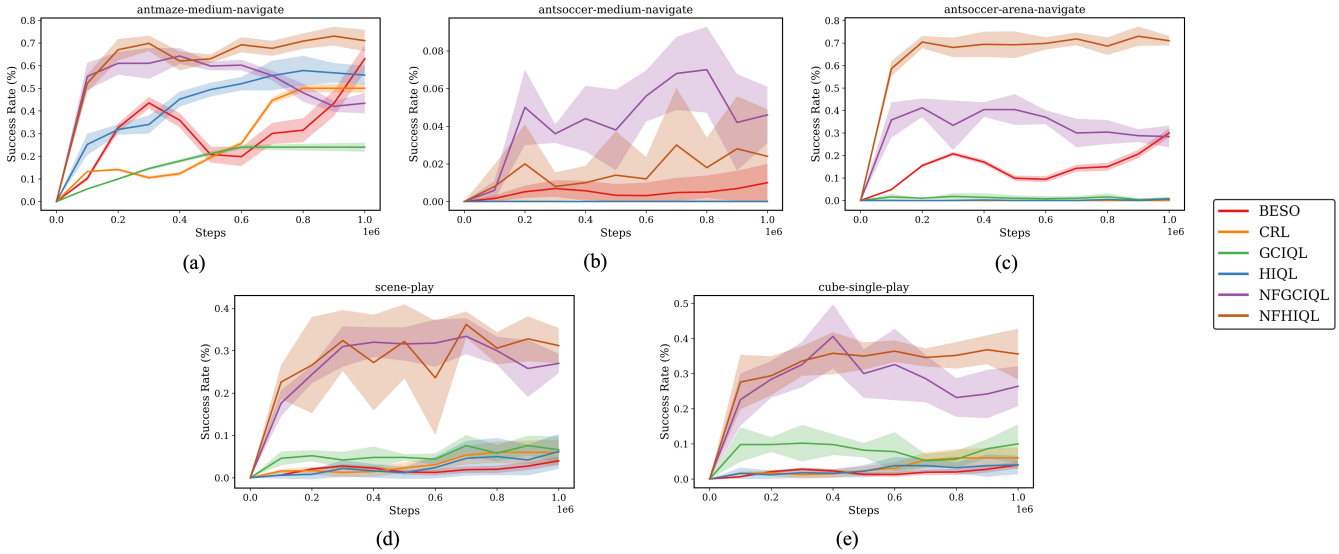

Fig. 2: Success rate (%) across training steps on OGBench environments. NF-HIQL consistently outperforms baselines, showing faster convergence and higher final success rates, particularly in complex manipulation tasks (`cube-single-play`, `scene-play`) and multi-agent soccer settings.

Environment	High-only	Low-only	Both
antmaze-med-nav	90 ±3	88 ±2	**95** ±2
antsoccer-arena-nav	60 ±3	52 ±4	**73** ±1
cube-single-play	28 ±3	32 ±4	**37** ±2
scene-play	34 ±3	30 ±4	**40** ±3

TABLE III: Ablation: normalizing flow at high-level only, low-level only, or both levels (100% dataset). Using flows at both levels consistently yields the best performance.

Environment	L=2	L=4	L=6	L=8
antmaze-med-nav	87 ±3	**95** ±2	94 ±2	92 ±3
antsoccer-arena-nav	55 ±4	**73** ±1	71 ±2	68 ±3
cube-single-play	25 ±4	**37** ±2	35 ±3	33 ±3

TABLE IV: Ablation: effect of flow depth (number of coupling layers L). Performance peaks around L=4 and degrades slightly with deeper flows, likely due to increased optimization difficulty.

VI. EXPERIMENTS

To validate NF-HIQL in real-world conditions, we deployed it on an Elephant Robotics 6-DOF myCobot280 arm with an adaptive gripper, running on a Jetson Nano AI board (Fig. 3). Two pick-and-place scenarios of increasing difficulty were tested: two-object and three-object sequential manipulation.

The algorithm was trained under two configurations: 3000 and 1500 offline samples. In all settings, the robot achieved 100% task success. With full data, average end-effector-to-target error stayed below 1.5 cm; with 50% data, errors rose modestly to ∼2.3 cm (two objects) and ∼2.8 cm (three objects), confirming robustness to reduced training data.

VII. CONCLUSION

This work introduced NF-HIQL, a normalizing flow–based extension of HIQL that employs expressive multimodal policies at both hierarchical levels. By replacing Gaussian policies with tractable flows, NF-HIQL enhances expressivity while maintaining stability through KL-divergence bounds and PAC-style sample efficiency guarantees. Experiments on OGBench show NF-HIQL consistently outperforms prior goal-conditioned and hierarchical baselines, including diffusion-based methods like BESO, especially in data-limited regimes. Notably, NF-HIQL trained with only

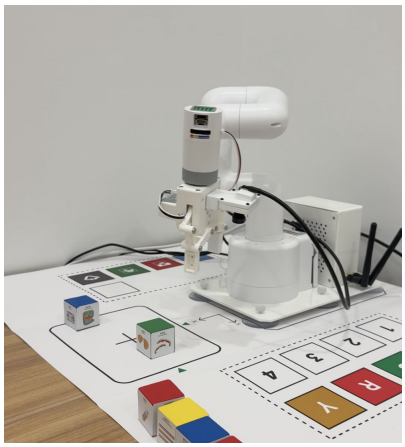

Fig. 3: Experimental setup with the 6 DOF myCobot 280 arm.

50% of the data matches or surpasses full-dataset baselines across navigation, ball-dribbling, and manipulation tasks, demonstrating strong robustness and efficiency.

Beyond simulation, we validated NF-HIQL on a real-world robotic platform using the myCobot280 arm, where the policy reliably executed multi-object pick-and-place tasks. Even with limited offline data, the algorithm achieved 100% task success with only modest increases in placement error, further underscoring its practical applicability in resource-constrained, real-world settings.

Overall, these results establish NF-HIQL as a scalable, data-efficient approach to HGCRL. By combining the theoretical guarantees of HIQL with the representational power of normalizing flows, NF-HIQL provides a compelling framework for robust decision-making in both offline and real-world scenarios. Future directions include extending this framework to vision-based inputs, integrating with model-based components for planning, and applying it to more complex multi-agent and long-horizon robotic systems.

REFERENCES

[1] C. M. Wu, B. Meder, and E. Schulz, "Unifying principles of generalization: past, present, and future," *Annual Review of Psychology*, vol. 76, 2024.

[2] G. B. Margolis, G. Yang, K. Paigwar, T. Chen, and P. Agrawal, "Rapid locomotion via reinforcement learning," *The International Journal of Robotics Research*, vol. 43, no. 4, pp. 572–587, 2024.

[3] T. Lin, K. Sachdev, L. Fan, J. Malik, and Y. Zhu, "Sim-to-real reinforcement learning for vision-based dexterous manipulation on humanoids," *arXiv preprint arXiv:2502.20396*, 2025.

[4] Z. Hao, G. Chen, Z. Huang, Q. Jia, Y. Liu, and Z. Yao, "Coordinated transportation of dual-arm robot based on deep reinforcement learning," in *2024 IEEE 19th Conference on Industrial Electronics and Applications (ICIEA)*. IEEE, 2024, pp. 1–6.

[5] X. Gong, F. Dawei, K. Xu, B. Ding, and H. Wang, "Goal-conditioned on-policy reinforcement learning," *Advances in neural information processing systems*, vol. 37, pp. 45 975–46 001, 2024.

[6] J. Li, C. Tang, M. Tomizuka, and W. Zhan, "Hierarchical planning through goal-conditioned offline reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 216–10 223, 2022.

[7] L. Blondé and A. Kalousis, "Sample-efficient imitation learning via generative adversarial nets," in *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, 2019, pp. 3138–3148.

[8] S. Mei and Y. Wu, "Deep networks as denoising algorithms: Sample-efficient learning of diffusion models in high-dimensional graphical models," *IEEE Transactions on Information Theory*, 2025.

[9] H. Cao, C. Tan, Z. Gao, Y. Xu, G. Chen, P.-A. Heng, and S. Z. Li, "A survey on generative diffusion models," *IEEE transactions on knowledge and data engineering*, vol. 36, no. 7, pp. 2814–2830, 2024.

[10] J. Xiong, G. Liu, L. Huang, C. Wu, T. Wu, Y. Mu, Y. Yao, H. Shen, Z. Wan, J. Huang *et al.*, "Autoregressive models in vision: A survey," *arXiv preprint arXiv:2411.05902*, 2024.

[11] J. Choi, S. Byeon, and I. Hwang, "Data-driven closed-loop reachability analysis for nonlinear human-in-the-loop systems using gaussian mixture model," *IEEE Transactions on Control Systems Technology*, 2024.

[12] S. Park, D. Ghosh, B. Eysenbach, and S. Levine, "Hiql: Offline goal-conditioned rl with latent states as actions," *Advances in Neural Information Processing Systems*, vol. 36, pp. 34 866–34 891, 2023.

[13] I. Kobyzev, S. J. Prince, and M. A. Brubaker, "Normalizing flows: An introduction and review of current methods," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 11, pp. 3964–3979, 2020.

[14] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real nvp," *arXiv preprint arXiv:1605.08803*, 2016.

[15] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, O. Pieter Abbeel, and W. Zaremba, "Hindsight experience replay," *Advances in neural information processing systems*, vol. 30, 2017.

[16] D. Yang, H. Zhang, X. Lan, and J. Ding, "Density-based curriculum for multi-goal reinforcement learning with sparse rewards," *arXiv preprint arXiv:2109.08903*, 2021.

[17] S. Nair, S. Savarese, and C. Finn, "Goal-aware prediction: Learning to model what matters," in *International Conference on Machine Learning*. PMLR, 2020, pp. 7207–7219.

[18] H. Charlesworth and G. Montana, "Plangan: Model-based planning with sparse rewards and multiple goals," *Advances in Neural Information Processing Systems*, vol. 33, pp. 8532–8542, 2020.

[19] M. Zhu, M. Liu, J. Shen, Z. Zhang, S. Chen, W. Zhang, D. Ye, Y. Yu, Q. Fu, and W. Yang, "Mapgo: Model-assisted policy optimization for goal-oriented tasks," *arXiv preprint arXiv:2105.06350*, 2021.

[20] T. D. Kulkarni, K. Narasimhan, A. Saeedi, and J. Tenenbaum, "Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation," *Advances in neural information processing systems*, vol. 29, 2016.

[21] O. Nachum, S. Gu, H. Lee, and S. Levine, "Near-optimal representation learning for hierarchical reinforcement learning," *arXiv preprint arXiv:1810.01257*, 2018.

[22] A. Robert, C. Pike-Burke, and A. A. Faisal, "Sample complexity of goal-conditioned hierarchical reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 36, pp. 62 696–62 712, 2023.

[23] V. Jain and S. Ravanbakhsh, "Learning to reach goals via diffusion," *arXiv preprint arXiv:2310.02505*, 2023.

[24] K. Madan, A. Zhan, A. Lamb, E. Bengio, L. Pan, G. Berseth, and Y. Bengio, "Goal2flownet: Learning diverse policy covers using gflownets for goal-conditioned rl."

[25] R. Ghugare and B. Eysenbach, "Normalizing flows are capable models for rl," *arXiv preprint arXiv:2505.23527*, 2025.

[26] B. Mazoure, T. Doan, A. Durand, J. Pineau, and R. D. Hjelm, "Leveraging exploration in off-policy algorithms via normalizing flows," in *Conference on Robot Learning*. PMLR, 2020, pp. 430–444.

[27] M. Reuss, M. Li, X. Jia, and R. Lioutikov, "Goal-conditioned imitation learning using score-based diffusion policies," *arXiv preprint arXiv:2304.02532*, 2023.

[28] R. Bellman, "A markovian decision process," *Journal of mathematics and mechanics*, pp. 679–684, 1957.

[29] S. Park, K. Frans, B. Eysenbach, and S. Levine, "Ogbench: Benchmarking offline goal-conditioned rl," *arXiv preprint arXiv:2410.20092*, 2024.

APPENDIX

A. KL Bound for Advantage-Weighted RealNVP Policies

Lemma 1 (RealNVP Lower Bound). *Fix $s \in \mathcal{S}$ and let $\pi_\theta(a \mid s)$ be a RealNVP density on a with L inverse coupling layers. Therefore, $u = x^{(L)} = f_\theta^{-1}(a; s)$ and*

$$\log \pi_\theta(a \mid s) = \log p_z(u) + \log \Big| \det \frac{\partial u}{\partial a} \Big|, \qquad p_z(u) = \mathcal{N}(0, I). \tag{12}$$

Assume (i) bounded actions: $\|a\|_2 \leq A_{\max}$*; (ii) per-layer bounds:* $\|s_\ell(\cdot)\|_\infty \leq S_\ell$, $\|t_\ell(\cdot)\|_2 \leq T_\ell$*; and (iii)* $d_\ell := |I_\ell|$ *scaled coordinates. Then, for all such* a*,*

$$\log \pi_\theta(a \mid s) \geq -B, \tag{13}$$

$$B := \frac{d}{2} \log(2\pi) \; + \; \frac{1}{2} U_{\max}^2 \; + \; \sum_{\ell=1}^{L} d_\ell S_\ell, \tag{14}$$

$$U_{\max} := \exp\Big(\sum_{\ell=1}^{L} S_\ell \Big) \Big(A_{\max} + \sum_{\ell=1}^{L} T_\ell \Big). \tag{15}$$

Proof. Let $x^{(0)} = a$ and $x^{(L)} = u$. For any diagonal matrix D, $\|Dv\|_2 \leq \|D\|_2 \, \|v\|_2$ and for $D = \mathrm{Diag}(e^{-s_\ell})$ we have $\|D\|_2 = \max_j e^{-s_{\ell,j}} \leq e^{S_\ell}$ by $\|s_\ell\|_\infty \leq S_\ell$. Thus, using the inverse update and the triangle inequality,

$$\|x^{(\ell+1)}\|_2 \leq e^{S_\ell} \|x^{(\ell)}\|_2 + e^{S_\ell} T_\ell. \tag{16}$$

Iterating this inequality for $\ell = 0, \ldots, L-1$ gives

$$\|u\|_2 = \|x^{(L)}\|_2 \leq \exp\Big(\sum_{\ell=1}^{L} S_\ell \Big) \Big(\|a\|_2 + \sum_{\ell=1}^{L} T_\ell \Big) \leq U_{\max}. \tag{17}$$

For the Gaussian base,

$$-\log p_z(u) = \frac{d}{2} \log(2\pi) + \frac{1}{2} \|u\|_2^2 \leq \frac{d}{2} \log(2\pi) + \frac{1}{2} U_{\max}^2. \tag{18}$$

The inverse Jacobian of a RealNVP layer is block-triangular with diagonal $\mathrm{Diag}(e^{-s_\ell})$ on the scaled block, hence

$$\log \Big| \det \frac{\partial x^{(\ell+1)}}{\partial x^{(\ell)}} \Big| = - \sum_{j \in I_\ell} s_{\ell,j}(x^{(\ell)}_{J_\ell}) \geq -d_\ell S_\ell. \tag{19}$$

Summing over layers yields

$$\log \Big| \det \frac{\partial u}{\partial a} \Big| = \sum_{\ell=0}^{L-1} \log \Big| \det \frac{\partial x^{(\ell+1)}}{\partial x^{(\ell)}} \Big| \geq - \sum_{\ell=1}^{L} d_\ell S_\ell. \tag{20}$$

Combining the base and Jacobian bounds gives

$$\log \pi_\theta(a \mid s) = \log p_z(u) + \log \Big| \det \frac{\partial u}{\partial a} \Big| \geq -\Big[\frac{d}{2} \log(2\pi) + \frac{1}{2} U_{\max}^2 \Big] - \sum_{\ell=1}^{L} d_\ell S_\ell \tag{21}$$

$$= -B. \tag{22}$$

□

Lemma 2 (KL Bound with Behavior Density Cap)**.** *Let* $\pi^b(\cdot \mid s)$ *be a behavior policy with* $\pi^b(a \mid s) \leq M < \infty$ *for all* a, s*. If* $\pi_\theta(\cdot \mid s)$ *satisfies Lemma 1 with constant* B*, then for every* s*,*

$$\mathrm{KL}(\pi^b(\cdot \mid s) \,\|\, \pi_\theta(\cdot \mid s)) = \underbrace{\int \pi^b(a \mid s)[-\log \pi_\theta(a \mid s)] \, da}_{\mathcal{H}(\pi^b, \pi_\theta)} - \underbrace{\int \pi^b(a \mid s)[-\log \pi^b(a \mid s)] \, da}_{\mathcal{H}(\pi^b)} \leq B + \log M. \tag{23}$$

Proof. By Lemma 1, $-\log \pi_\theta(a \mid s) \leq B$ on the support of $\pi^b(\cdot \mid s)$, so $\mathcal{H}(\pi^b, \pi_\theta) \leq B$. Since $\pi^b(a \mid s) \leq M$ a.e., $\log \pi^b(a \mid s) \leq \log M$ a.e., hence

$$\begin{aligned} \mathcal{H}(\pi^b) &= -\int \pi^b \log \pi^b \, da \\ &\geq -\int \pi^b \log M \, da \\ &= -\log M. \end{aligned} \tag{24}$$

Therefore $\mathrm{KL}(\pi^b \| \pi_\theta) \leq B - (-\log M) = B + \log M$. □

Lemma 3 (RealNVP Upper Bound on Log-Density)**.** *Under the same assumptions as Lemma 1, and additionally assuming* $\|s_\ell(\cdot)\|_\infty \geq s_{\min} > 0$ *(i.e., the scale networks are bounded away from zero), the log-density is also upper-bounded:*

$$\log \pi_\theta(a \mid s) \leq B^+, \tag{25}$$

where $B^+ := \sum_{\ell=1}^{L} d_\ell S_\ell$ *depends on the same architectural constants. Together with Lemma 1, this yields* $|\log \pi_\theta(a \mid s)| \leq \max(B, B^+)$*, establishing the two-sided boundedness required by Assumption (A4) in the sample efficiency proof.*

Proof. The Gaussian base density satisfies $\log p_z(u) \leq 0$ for all u (since p_z is a standard normal with maximum density $(2\pi)^{-d/2} \leq 1$ for $d \geq 1$). The log Jacobian determinant is $\sum_{\ell=1}^{L} \sum_{j \in I_\ell} (-s_{\ell,j})$. Since $|s_{\ell,j}| \leq S_\ell$, each term $-s_{\ell,j} \leq S_\ell$, so $\log|\det(\partial u / \partial a)| \leq \sum_{\ell=1}^{L} d_\ell S_\ell$. Combining: $\log \pi_\theta(a \mid s) = \log p_z(u) + \log|\det(\partial u / \partial a)| \leq 0 + \sum_{\ell=1}^{L} d_\ell S_\ell = B^+$. □

Remark on B and network complexity. B grows with the number of coupling layers L and the scale/translation network bounds S_ℓ, T_ℓ. Deeper or wider RealNVP networks increase B, loosening the KL bound—reflecting an expressivity–stability trade-off. In practice, bounded activations (e.g., `tanh`) keep B moderate. The bound depends only on architectural constants, not on training data or loss.

B. Sample Efficiency for HIQL with Flow Policies

For each level $L \in \{h, \ell\}$, let V be a shared value function used to compute per-level advantages $A_L(a, s)$. Define the advantage-weights as $w_L(a, s) = e^{\beta A_L(a,s)}$. Given an offline dataset inducing a distribution $d_L^\mu(s, a)$ at each level, the training objective is $\mathcal{J}_L(\pi) = \mathbb{E}_{(s,a) \sim d_L^\mu}[w_L(a, s) \log \pi(a \mid s)]$, under the following assumptions.

- **Bounded rewards:** $|R(s, g)| \leq R_{\max}$ and $\gamma \in (0, 1)$.
- **Concentrability (coverage):** For each level L, the reference state occupancy $d_L^\star$ is continuous w.r.t. the

dataset occupancy d_L^μ, with density ratio bounded by $C_L \geq 1$:

$$\sup_s \frac{d_L^\star(s)}{d_L^\mu(s)} \leq C_L. \tag{26}$$

- **Weight control:** Weights are bounded, either because of clipping: $0 \leq w_L(a,s) \leq W_{\max}$ or A_L is bounded.
- **Policy class capacity & bounded loss:** For each L, the log-density class $\mathcal{F}_L = \{(s,a) \mapsto \log \pi(a \mid s) : \pi \in \Pi_L\}$ has finite Rademacher complexity $\mathfrak{R}_{n_L}(\mathcal{F}_L)$, and $|\log \pi(a \mid s)| \leq B_L$. A bound on per-level *environment* advantages is also assumed: $|A^{\pi_L}(s,a)| \leq A_{\max,L}$.

1) KL reduction.: For each s, define

$$\begin{aligned} Z_L(s) &= \int w_L(a,s)\, p_{\text{data}}(a \mid s)\, da, \\ q_L(a \mid s) &= \frac{w_L(a,s)\, p_{\text{data}}(a \mid s)}{Z_L(s)}. \end{aligned} \tag{27}$$

Then

$$\begin{aligned} \mathcal{J}_L(\pi \mid s) &= \int w_L(a,s)\, p_{\text{data}}(a \mid s)\, \log \pi(a \mid s)\, da \\ &= Z_L(s)\, \mathbb{E}_{q_L}[\log \pi]. \end{aligned} \tag{28}$$

Adding and subtracting $\mathbb{E}_{q_L} \log q_L$ gives

$$\mathcal{J}_L(\pi \mid s) = Z_L(s) \Big(\mathbb{E}_{q_L} \log q_L - \mathrm{KL}(q_L \| \pi) \Big). \tag{29}$$

Thus

$$\sup_{\pi'} \mathcal{J}_L(\pi' \mid s) - \mathcal{J}_L(\pi \mid s) = Z_L(s)\, \mathrm{KL}(q_L \| \pi). \tag{30}$$

Averaging over $s \sim d_L^\mu$,

$$\sup_{\pi'} \mathcal{J}_L(\pi') - \mathcal{J}_L(\pi) = \mathbb{E}_{s \sim d_L^\mu}[\, Z_L(s)\, \mathrm{KL}(q_L \| \pi)\,]. \tag{31}$$

2) Weighted ERM Generalization Bounds: Let $\ell_\pi(a,s) = -\log \pi(a \mid s) \in [0, B_L]$ with $0 \leq w \leq W_{\max}$. Define

$$L(\pi) := \mathbb{E}[w\, \ell_\pi], \qquad \widehat{L}_n(\pi) := \frac{1}{n} \sum_{i=1}^n w_i\, \ell_\pi(a_i, s_i). \tag{32}$$

By symmetrization and Hoeffding's inequality, with probability $\geq 1 - \delta$,

$$\sup_{\pi \in \Pi_L} |L(\pi) - \widehat{L}_n(\pi)| \leq 2\, \mathfrak{R}_n(\mathcal{G}_L) + W_{\max} B_L \sqrt{\frac{\log(1/\delta)}{2n}}, \tag{33}$$

where $\mathcal{G}_L = \{(s,a) \mapsto w(s,a)\, \ell_\pi(s,a) : \pi \in \Pi_L\}$ and

$$\mathfrak{R}_n(\mathcal{G}_L) := \mathbb{E}_{\boldsymbol{z}, \boldsymbol{\sigma}} \left[\sup_{\pi \in \Pi_L} \frac{1}{n} \sum_{i=1}^n \sigma_i\, w_i\, \ell_\pi(a_i, s_i) \right], \tag{34}$$

with σ_i i.i.d. Rademacher variables.

3) Multiplier contraction (bounded weights).: For $\mathcal{H} = \{\ell_\pi : \pi \in \Pi_L\}$ and $\mathcal{G}_L = \{w\, \ell_\pi : \ell_\pi \in \mathcal{H}\}$ with $0 \leq w \leq W_{\max}$, the Ledoux–Talagrand contraction inequality with $\phi_i(t) = \alpha_i t$, $\alpha_i = w_i / W_{\max} \in [0,1]$ and for fixed sample $(z_i) = (s_i, a_i)$, write

$$\begin{aligned} \mathfrak{R}_n(\mathcal{G}_L \mid \boldsymbol{z}) &= \mathbb{E}_{\boldsymbol{\sigma}} \left[\sup_\pi \frac{1}{n} \sum_{i=1}^n \sigma_i w_i\, \ell_\pi(z_i) \right] \\ &= W_{\max}\, \mathbb{E}_{\boldsymbol{\sigma}} \left[\sup_\pi \frac{1}{n} \sum_{i=1}^n \sigma_i \alpha_i\, \ell_\pi(z_i) \right], \end{aligned} \tag{35}$$

The contraction inequality gives -

$$\mathbb{E}_{\boldsymbol{\sigma}} \left[\sup_\pi \frac{1}{n} \sum_{i=1}^n \sigma_i \phi_i(\ell_\pi(z_i)) \right] \leq \mathbb{E}_{\boldsymbol{\sigma}} \left[\sup_\pi \frac{1}{n} \sum_{i=1}^n \sigma_i \ell_\pi(z_i) \right]. \tag{36}$$

Taking expectation over $\boldsymbol{z}$ yields $\mathfrak{R}_n(\mathcal{G}_L) \leq W_{\max}\, \mathfrak{R}_n(\mathcal{H})$. Therefore, with probability $\geq 1 - \delta$,

$$\begin{aligned} \sup_{\pi \in \Pi_L} |L(\pi) - \widehat{L}_n(\pi)| \leq 2 W_{\max}\, \mathfrak{R}_n(\mathcal{F}_L) + \\ W_{\max} B_L \sqrt{\frac{\log(1/\delta)}{2n}}. \end{aligned} \tag{37}$$

4) ERM ⇒ KL control.: Let $\hat{\pi}_L = \arg\max_{\pi \in \Pi_L} \widehat{\mathcal{J}}_L(\pi)$. Since

$$\widehat{\mathcal{J}}_L(q_L) - \widehat{\mathcal{J}}_L(\hat{\pi}_L) \leq 0, \tag{38}$$

we obtain

$$\mathcal{J}_L(q_L) - \mathcal{J}_L(\hat{\pi}_L) \leq 2 \sup_\pi |\mathcal{J}_L(\pi) - \widehat{\mathcal{J}}_L(\pi)|. \tag{39}$$

From the KL reduction step,

$$\mathcal{J}_L(q_L) - \mathcal{J}_L(\hat{\pi}_L) = \mathbb{E}_x \Big[Z_L(s)\, \mathrm{KL}(q_L \| \hat{\pi}_L) \Big]. \tag{40}$$

Using $Z_L(s) \leq W_{\max}$ and the ERM bound,

$$\mathbb{E}_s \mathrm{KL}(q_L(\cdot \mid s) \,\|\, \hat{\pi}_L(\cdot \mid s)) \leq 4\, \mathfrak{R}_n(\mathcal{F}_L) + 2 B_L \sqrt{\frac{\log(1/\delta)}{2n}}. \tag{41}$$

5) Return gap from KL.: For any level L with effective timescale H_L and per-level policies $\pi_L^\star, \pi_L$, the performance-difference lemma gives

$$J(\pi_L^\star) - J(\pi_L) = \frac{1}{1-\gamma}\, \mathbb{E}_{s \sim d^{\pi_L^\star}, a \sim \pi_L^\star}[A^{\pi_L}(s,a)]. \tag{42}$$

Since $|A^{\pi_L}| \leq A_{\max,L}$ and $\mathrm{TV}(p,q) \leq \sqrt{\frac{1}{2} \mathrm{KL}(p \| q)}$,

$$\mathbb{E}_{a \sim \pi_L^\star}[A^{\pi_L}(s,a)] \leq A_{\max,L} \sqrt{\frac{1}{2} \mathrm{KL}(\pi_L^\star \| \pi_L)}. \tag{43}$$

Applying Jensen to $\sqrt{\cdot}$ and concentrability,

$$J(\pi_L^\star) - J(\pi_L) \leq \frac{H_L A_{\max,L}}{1-\gamma} \sqrt{\frac{C_L}{2}\, \mathbb{E}_{s \sim d_L^\mu} \mathrm{KL}(\pi_L^\star \| \pi_L)}. \tag{44}$$

6) Main Theorem:

Lemma 4 (PAC-style sample efficiency with explicit constants)**.** *Let $\hat{\pi}_h, \hat{\pi}_\ell$ be the learned per-level policies via advantage-weighted MLE with conditional flows. Under Assumptions 1–4, with probability at least $1 - \delta$,*

$$\begin{aligned} &J(\pi^\star) - J(\hat{\pi}_{h,\ell}) \\ &\leq \frac{H_h A_{\max,h}}{1-\gamma} \sqrt{\frac{C_h}{2} \left(4\, \mathfrak{R}_{n_h}(\mathcal{F}_h) + 2 B_h \sqrt{\frac{\log(2/\delta)}{2 n_h}} \right)} \\ &\quad + \frac{H_\ell A_{\max,\ell}}{1-\gamma} \sqrt{\frac{C_\ell}{2} \left(4\, \mathfrak{R}_{n_\ell}(\mathcal{F}_\ell) + 2 B_\ell \sqrt{\frac{\log(2/\delta)}{2 n_\ell}} \right)} \\ &\quad + \varepsilon_V, \end{aligned} \tag{45}$$

where ε_V is the uniform value-estimation error used to compute A_L.

Proof. Apply Theory in VII-B.4 to bound $\mathbb{E}\mathrm{KL}(q_L \| \hat{\pi}_L)$, then theory in VII-B.5 with $\pi_L^\star := q_L$ and $\pi_L := \hat{\pi}_L$ for each level, and sum the two contributions. Add ε_V for the value approximation. □