DOUBLY DUAL-WEIGHTED FLOW MATCHING FOR HIERARCHICAL OFFLINE GOAL-CONDITIONED RL

Anonymous authors
Paper under double-blind review

ABSTRACT

Offline goal-conditioned reinforcement learning (GCRL) learns goal-reaching policies from static datasets without further environment interaction. We propose *Doubly Dual-Weighted Flow Matching* (D^2FM), a hierarchical goal-conditioned flow policy that mitigates distant goals by decomposing long-horizon goal reaching into a high-level module that proposes nearer subgoals and a low-level flow module that generates actions conditioned on those subgoals. To mitigate sparse rewards, we introduce a doubly dual-weight mechanism that derives two optimal correction ratios from a single reward-tilted target distribution: a state–action weight ratio, obtained from a primal–dual Bellman-flow program for low-level policy extraction; and a state weight ratio, recovered through an auxiliary off-policy evaluation objective, that reweights high-level subgoal selection. On OGBench, a benchmark covering navigation, manipulation, and combinatorial-puzzle tasks, D^2FM achieves higher average success rates than the evaluated GCRL baselines on long-horizon goal-reaching tasks.

1 INTRODUCTION

Offline goal-conditioned reinforcement learning (GCRL) aims to learn a goal-conditioned policy that reaches an arbitrary designated goal from a static offline dataset. Typically, the policy is trained with a behavior-cloning objective, and advantage-weighted regression (AWR) based on the state-action value function is often used to distinguish the importance of each sample (Yang et al., 2022b; 2023; Hejna et al., 2023). Practically, the performance of AWR-based policies usually degrades as the task horizon grows (Springenberg et al., 2024; Park et al., 2025b). Note that, for long-horizon tasks, the successful trajectories in the pre-collected offline dataset are usually mixed with a large fraction of failed, suboptimal samples (Kumar et al., 2022; Kim et al., 2024), since distant goals are harder to reach. While more precise importance reweighting is required for the skewed suboptimal long-horizon dataset, the sparse long-horizon rewards yield imprecise value functions with low signal-to-noise ratio, and the advantage gaps used to rank the samples shrink as goals become more distant (Xu et al., 2022; Park et al., 2023), making value-based credit assignment unreliable. Moreover, the policy must stitch together individual segments among trajectories associated with different goals (Gupta et al., 2019; Hoang et al., 2021; Zhang et al., 2021), a capability that direct behavior cloning on a suboptimal dataset does not provide.

Prior work improves performance on long-horizon tasks along two complementary axes (Table 1). The first is *DIstribution Correction Estimation (DICE)*, which directly attacks the noisy-advantage problem by reweighting samples with a dual-derived state–action ratio. Specifically, DICE-style methods such as GoFAR (Ma et al., 2022b) and SMORe (Sikchi et al., 2023) replace the advantage weight with a primal–dual occupancy ratio that is more principled and with lower variance than exponentiated state–action advantage weights. The second is *hierarchical decomposition*, which splits goal reaching into high-level subgoal selection in the state space and low-level control that generates actions conditioned on those subgoals, thereby shortening the effective horizon (Chane-Sane et al., 2021; Li et al., 2022; Park et al., 2023). Recently, the goal-conditioned flow actor and classifier-free guidance are also integrated into this hierarchy to further enhance stitching ability of hierarchical policies (Janner et al., 2022; Kim et al., 2024; Frans et al., 2025). A natural question then arises as follows.

Does there exist an effective distribution-correction estimator for hierarchical policies?

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.

The extension is not as direct as it seems. Existing DICE methods (Ma et al., 2022b; Sikchi et al., 2023) only recover a state–action ratio suitable for the low-level control via the Bellman flow, whereas closing the gap requires a weighting mechanism that operates at *both* levels of the hierarchy. The *state-level* ratio required for action-free high-level subgoal selection cannot be directly recovered by current DICE-style methods.

We introduce *Doubly Dual-Weighted Flow Matching* (D^2FM), which closes this gap with two coupled weighting objectives derived from a single reward-tilted target distribution. At the state–action level, we formulate occupancy correction as a primal–dual Bellman-flow problem anchored to the relabeled offline occupancy d^O_{SA}, yielding the optimal data-supported ratio $w^*_{SA} = d^*_{SA}/d^O_{SA}$ for low-level policy extraction. Building on w^*_{SA}, we construct an auxiliary *off-policy evaluation* objective that recovers the marginal state ratio $w^*_S = d^*_S/d^O_S$, which prior DICE-style methods neglect. The two ratios are then applied in the training procedure of the low-level/high-level flow policies. Specifically, w^*_{SA} directly reweights low-level conditional flow matching for action generation, while w^*_S is used to correct the action-free state value function learning through a weighted temporal difference learning. The resulting corrected value then drives advantage-weighted high-level subgoal selection, so that subgoal advantages are estimated under the optimal state distribution rather than the raw offline marginal. For efficient low-level control, we adopt the one-step policy extraction scheme of FQL, using our occupancy-weighted flow as the teacher and a critic trained from the corrected value function. The resulting distilled policy is a one-step generator that retains multimodal action coverage while eliminating iterative action generation at inference time. To summarize, our main contributions in this paper are listed as follows.

- **Data-supported DICE objective for low-level extraction.** We construct a data-supported, reward-tilted target $p_\alpha(s, a; g) \propto d^O_{SA}(s, a; g) \exp(r(s; g)/\alpha)$, and propose to learn the distribution correction estimator through offline occupancy matching against tilted target p_α instead of the vanilla reward seeking occupancy proportional to $\exp(r(s; g))$. The resulting optimal state–action ratio $w^*_{SA} = d^*_{SA}/d^O_{SA}$ provides data-supported weights for policy extraction that balance reward maximization against deviations from the offline occupancy, while the reward-tilted formulation eliminates the need for an auxiliary discriminator to estimate the target-to-data density ratio. The correction ratio w^*_{SA} is then directly used in the weighted behavior-cloning objective to extract the low-level conditional flow policy.
- **Auxiliary OPE for state correction ratio.** Building on w^*_{SA}, we construct an auxiliary off-policy evaluation (OPE) program whose Bellman-flow constraint is induced by the corrected state–action occupancy. By investigating the Fenchel-Rockafellar duality of the OPE problem, we derive a simple unconstrained dual problem to recover the marginal state correction ratio $w^*_S = d^*_S/d^O_S$ that prior DICE-style methods fail to achieve. Instead of directly correcting the high-level subgoal selection policy training in a behavior-clone manner, we use w^*_S to correct action-free state value learning through a weighted temporal-difference update. The resulting corrected value V_ψ can not only be used to construct the corrected advantage-weighted high-level subgoal selection objective, but also to obtain an action critic for low-level action policy training through the expectile regression.

Empirically, experiments on OGBench (Park et al., 2025a), a benchmark specifically designed for offline GCRL, demonstrate that D^2FM achieves the strongest average performance among the evaluated methods across diverse environments and task configurations.

2 Preliminaries

Offline GCRL. We consider an infinite-horizon goal-conditioned MDP (Puterman, 2014) with state space $\mathcal{S}$, action space $\mathcal{A}$, transition kernel T, initial distribution μ_0, discount factor $\gamma \in (0, 1)$, and goal space G. The reward $r(s; g)$[1] and policy $\pi(a \mid s, g)$ are conditioned on a goal $g \sim p(g)$, and the objective is

$$J(\pi) := \mathbb{E}_{g\sim p(g), s_0\sim\mu_0, a_t\sim\pi(\cdot|s_t,g), s_{t+1}\sim T(\cdot|s_t,a_t)} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t; g) \right]. \tag{1}$$

[1] In GCRL the reward customarily does not depend on the action.

Table 1: Offline GCRL methods categorized according to policy structure and weighting mechanism.

Hierarchy	Weighting	Policy class	Method
Flat	None	Gaussian	GCBC
	None	Flow	GCFBC
	AWR	Gaussian	WGCSL
	DICE	Gaussian	GoFAR
	CFG at test	Flow	CFGRL
Hierarchical	AWR (low,high)	Gaussian	HIQL
	CFG at test (low,high)	Flow	HCFGRL
	DICE (low), OPE (high)	Flow	**D^{2}FM**

We write the discounted state-action occupancy induced by π as $d^\pi_{SA}(s,a;g) := (1-\gamma)\sum_{t=0}^{\infty}\gamma^t \Pr\big(s_t = s,\, a_t = a \,\big|\, \pi, T, \mu_0, g\big)$, where the state marginal satisfies $d^\pi_S(s;g) = \sum_a d^\pi_{SA}(s,a;g)$ and $\pi(a \mid s,g) = d^\pi_{SA}(s,a;g)/d^\pi_S(s;g)$ holds for the policy. Any valid state-action occupancy must satisfy the Bellman-flow constraint

$$\sum_a d^\pi_{SA}(s,a;g) = (1-\gamma)\mu_0(s) + \gamma \sum_{\tilde{s},\tilde{a}} T(s \mid \tilde{s},\tilde{a})\, d^\pi_{SA}(\tilde{s},\tilde{a};g), \qquad \forall s \in \mathcal{S},\ g \in G. \tag{2}$$

In offline GCRL, the agent only has access to a static dataset $\mathcal{D}_O$ of logged trajectories. We denote its empirical goal-conditioned state-action occupancy by $d^O_{SA}(s,a;g)$ and its state marginal by $d^O_S(s;g)$.

DICE for offline GCRL. A line of offline GCRL work adapts DICE-style occupancy correction to goal reaching by viewing policy improvement as state-occupancy matching (Ma et al., 2022b; Zhu & Zhang, 2023; Sikchi et al., 2023). For a fixed goal g, a successful policy should place high occupancy on high-reward states, so the objective can be formulated as matching $d^\pi_S(s;g)$ to a reward-tilted target $p(s,a;g) \propto \exp(r(s;g))$. Because d^π_S cannot be sampled directly from static data, DICE-style methods optimize a surrogate over d^π_{SA} by anchoring a joint f-divergence to $p(s,a;g)$ and enforcing the bellman-flow constraint (2). The dual yields the state-action correction ratio, which is then used as an importance weight for behavior-cloning type policy extraction.

Hierarchy and flow policies. To mitigate the long-horizon goal-reaching challenge, HIQL (Park et al., 2023) decomposes goal-conditioned control into high-level subgoal selection and low-level action execution. It learns a goal-conditioned value function from offline data using implicit Q-learning (Kostrikov et al., 2022), and then extracts both high- and low-level policies via advantage-weighted regression. Recent flow-based methods (Park et al., 2025c; Frans et al., 2025) replace the unimodal Gaussian actors used in prior approaches with conditional flow-matching actors, improving policy expressiveness. Their behavior-cloning objective is

$$\mathcal{L}_{\text{Flow}}(\theta) = \mathbb{E}_{x^1 \sim \mathcal{D}^O,\, x^0 \sim \mathcal{N}(0,I_d),\, \sigma \sim \text{Unif}([0,1])}\big[\|v_\theta(\sigma, s, g, x^\sigma) - (x^1 - x^0)\|_2^2\big], \tag{3}$$

where $x^\sigma = (1-\sigma)x^0 + \sigma x^1$. The ODE solution at $\sigma = 1$ is denoted $\mu_\theta(s,g,z)$, which maps Gaussian noise z to an action in the action policy, or a subgoal for the high-level flow policy. To our knowledge, no prior work derives two dual weights for a hierarchical goal-conditioned policy. CORSDICE (Kim et al., 2025) recovers a state correction from a policy correction with respect to a given policy for cost evaluation in constrained offline RL, whereas we aim to recover the optimal state correction from a state–action occupancy correction for high-level value learning.

3 Method

Overview. We propose Doubly Dual-Weighted Flow Matching (D^{2}FM), a hierarchical flow policy with a dual-derived weight applied at each level of the hierarchy. Starting from a data-supported, reward-tilted target distribution, we show that matching to this target is equivalent to a dataset-regularized GCRL objective. We then apply DICE under the Bellman flow constraint to recover the optimal state-action ratio w^*_{SA}. Building on w^*_{SA}, we construct an auxiliary OPE program that recovers the optimal state-level ratio w^*_S. Then we apply these two ratios via a doubly dual-weighted mechanism, both for value-learning and policy extraction. For fast deployment, we further distill the low-level flow into a one-step generator guided by a weighted critic. Algorithm 1 (in Appendix F) summarizes the full procedure.

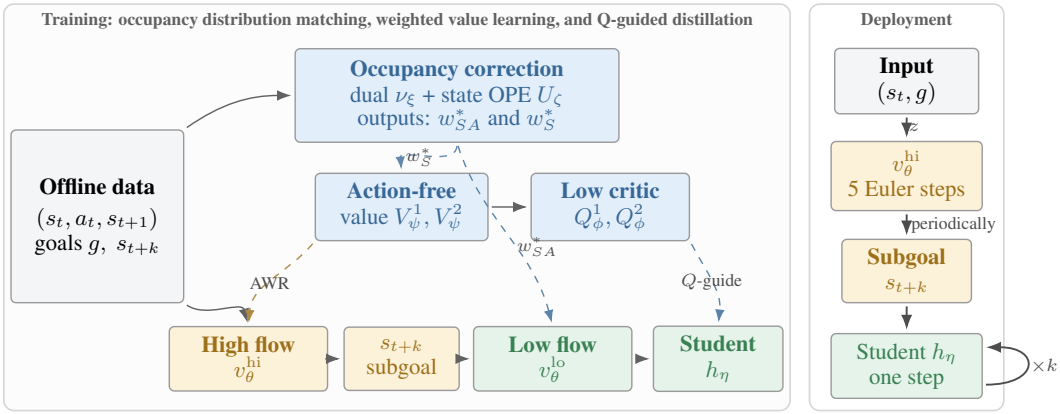


Figure 1: **D^{2}FM architecture.** D^{2}FM first solves an offline primal-dual problem that yields dual-level weights: the state-action ratio w^*_{SA} and the state ratio w^*_S. These weights drive a doubly dual-weight mechanism. Q_ϕ further guides the distillation of the fast one-step low-level policy h_η. *Deployment* (right panel): v^{hi}_θ is invoked periodically to propose a subgoal s_{t+k}, and h_η then runs in a k-step inner loop.

3.1 Dual-level weight derivation

Hierarchical policies for long-horizon tasks require weight signals at different levels: a state-action level weight for the low-level policy to generate actions and a state-level weight for the high-level policy to propose subgoals. Prior DICE methods recover only a state-action optimal occupancy ratio via the state-action level Bellman flow constraints; we build on this formulation by estimating the corresponding marginal state ratio to reweight action-free value learning that guides high-level subgoal selection. In this section, we derive both dual-ratios from a single data-supported reward-tilted target distribution, which are then applied at both levels of hierarchy (detailed in §3.2). From the state-occupancy view of offline GCRL (Lee et al., 2019; Ma et al., 2022a), prior work (Ma et al., 2022b; Zhu & Zhang, 2023) matches d^π_S to the reward-only target $p_R(s;g) \propto \exp(r(s;g))$. We instead match against a data-supported target that anchors the reward-tilt on the offline state distribution:

$$p_\alpha(s;g) \;\propto\; d^O_S(s;g)\,\exp\big(r(s;g)/\alpha\big), \tag{4}$$

where d^O_S denotes the empirical goal-conditioned state occupancy induced by offline transitions with goal relabeling, and $\alpha > 0$ is a temperature controlling how strongly reward reshapes d^O_S. Proposition 3.1 shows that matching d^π_S to p_α yields a tractable upper bound on the corresponding regularized goal-reaching objective.

Proposition 3.1. *Given any $r(s;g)$, for each goal in the support of $p(g)$, let p_α be the normalized distribution in (4), with normalizer $Z_\alpha(g)$. Then the following identity and lower bound hold for any f-divergence satisfying $D_f(P\|Q) \geq D_{\text{KL}}(P\|Q)$:*

$$\begin{aligned}
&-\alpha\mathbb{E}_{g\sim p(g)}\left[D_{\text{KL}}\big(d^\pi_S(\cdot;g)\|p_\alpha(\cdot;g)\big)\right] + C_\alpha \\
&\quad= (1-\gamma)J(\pi) - \alpha\mathbb{E}_{g\sim p(g)}\left[D_{\text{KL}}\big(d^\pi_S(\cdot;g)\|d^O_S(\cdot;g)\big)\right] \\
&\quad\geq (1-\gamma)J(\pi) - \alpha\mathbb{E}_{g\sim p(g)}\left[D_f\big(d^\pi_{SA}(\cdot,\cdot;g)\|d^O_{SA}(\cdot,\cdot;g)\big)\right],
\end{aligned} \tag{5}$$

where $C_\alpha \coloneqq \alpha\mathbb{E}_{g\sim p(g)}[\log Z_\alpha(g)]$ is constant with respect to π.

Remark 3.2. The identity in equation 5 shows that matching d^π_S to p_α is equivalent to maximizing a goal-reaching objective regularized by the state-marginal KL divergence. By anchoring p_α on d^O_S, this regularizer penalizes deviations from the offline state distribution during return maximization. The inequality in equation 5 then yields a state–action f-divergence-regularized lower-bound surrogate, which we optimize subject to the Bellman flow constraint (equation 2). Since $\log \frac{p_\alpha(s;g)}{d^O_S(s;g)} = \frac{r(s;g)}{\alpha} - \log Z_\alpha(g)$, the log density ratio is therefore directly available from rewards up to a goal-dependent constant, eliminating the need for an auxiliary discriminator to estimate it, as in Ma et al. (2022b); Jain & Unhelkar (2024). The proof is in Appendix L.2.

State–action ratio. Now, we focus on the following primal program to obtain the optimal d^*_{SA},

$$\begin{aligned} &\text{(P)} \quad \max_{d_{SA}(s,a;g)\geq 0} \mathbb{E}_{(s,g)\sim d(s,g)}[r(s;g)] - \alpha\, D_f(d_{SA}(s,a;g)\|d^O_{SA}(s,a;g)) \\ &\text{s.t.} \quad \textstyle\sum_a d_{SA}(s,a;g) = (1-\gamma)\mu_0(s) + \gamma \sum_{\tilde{s},\tilde{a}} T(s \mid \tilde{s},\tilde{a}) d_{SA}(\tilde{s},\tilde{a};g), \quad \forall s,g. \end{aligned} \tag{6}$$

While directly optimizing $D_{\mathrm{KL}}\big(d^\pi_S(s;g)\|p_\alpha(s;g)\big)$ to find d^*_{SA} is generally infeasible (Lee et al., 2021), the Bellman-flow constraint problem admits a feasible dual which can be optimized from offline samples without policy rollouts. The following proposition 3.3 gives the tractable dual convex optimization and derives a closed-form solution for the optimal state–action occupancy ratio via Lagrangian duality. The proof is in Appendix L.3.

Proposition 3.3 (Dual problem). *The dual of equation 6 is*

$$\text{(D)} \quad \min_{\nu(s;g)} (1-\gamma)\mathbb{E}_{s\sim\mu_0, g\sim p(g)}[\nu(s;g)] + \alpha \mathbb{E}_{(s,a,g)\sim d^O_{SA}}\left[f^*\left(\frac{\mathcal{T}\nu(s,a;g) - \nu(s;g)}{\alpha}\right)\right], \tag{7}$$

where f^* is the convex conjugate of f, and $\mathcal{T}\nu(s,a;g) = r(s;g) + \gamma \mathbb{E}_{s'\sim T(\cdot|s,a)}[\nu(s';g)]$. The optimal state–action occupancy ratio can be recovered from the optimal $\nu*$

$$w^*_{SA}(s,a;g) = \frac{d^*_{SA}(s,a;g)}{d^O_{SA}(s,a;g)} = \max\left(0, (f')^{-1}\left(\frac{\mathcal{T}\nu^*(s,a;g) - \nu^*(s;g)}{\alpha}\right)\right). \tag{8}$$

Why a single ratio cannot serve both levels. The high-level value loss in §3.2 is action-free and therefore needs the marginal state ratio $w_S = d^*_S/d^O_S$. This ratio satisfies $w_S(s;g) = \mathbb{E}_{a\sim d^O(\cdot|s,g)}[w_{SA}(s,a;g)]$ With the fitted w_{SA} held fixed, we estimate w_S through an auxiliary off-policy evaluation (OPE) (Uehara et al., 2022) formulation. Building on techniques from Nachum & Dai (2020), we formulate the following constrained optimization problem:

$$\begin{aligned} &\max_{d_S\geq 0} -h(d_S) \\ &\text{s.t. } d_S(s;g) = (1-\gamma)\mu_0(s) + \gamma \sum_{\bar{s},\bar{a}} d^O_{SA}(\bar{s},\bar{a};g)\, w^*_{SA}(\bar{s},\bar{a};g)\, T(s \mid \bar{s},\bar{a}), \ \forall s \in \mathcal{S}. \end{aligned} \tag{9}$$

The equality constraints already pin down d_S uniquely for given goal g; the regularizer h is included to make the dual derivation tractable. Common choices include $h(d_S) := 0$ or $h(d_S) := D_f(d_S\|d^O_S)$.

Proposition 3.4 (State-ratio dual). *Applying Fenchel-Rockafellar duality with $h(d_S) := D_f(d_S\|d^O_S)$ to equation 9 yields:*

$$\min_U \mathbb{E}_{s\sim d^O_S}\big[f^*\big(U(s;g)\big)\big] - (1-\gamma)\,\mathbb{E}_{s\sim\mu_0}[U(s;g)] - \gamma\,\mathbb{E}_{(s,a,s')\sim d^O}\big[w_{SA}(s,a;g)\cdot U(s';g)\big]. \tag{10}$$

The recovered state ratio is

$$w^*_S(s;g) = \max\big(0,\ (f')^{-1}(U^*(s;g))\big). \tag{11}$$

The derivation is in Appendix L.4. In practice, we parameterize the dual variables ν_ξ and U_ζ as neural networks and optimize equation 7 and equation 10 by stochastic gradient descent on offline samples. During each update of U_ζ, the estimated state–action ratio $\widehat{w}_{SA}$ is treated as a stop-gradient input. The learned state ratio $\widehat{w}_S$ is obtained from U_ζ via equation 11 and used to weight the value-learning objective. With finite data and neural function approximation, $\widehat{w}_S$ approximates the ratio induced by the current $\widehat{w}_{SA}$ through the state-level Bellman flow.

3.2 Doubly-Weighted Flow Matching

We now instantiate the learned state and state-action ratios, w_S and w_{SA}, in a hierarchical flow policy for long-horizon offline GCRL. The state ratio first reshapes value learning so that candidate subgoals are evaluated under the corrected occupancy; these values then guide a high-level subgoal flow, while the state-action ratio weights the low-level action flow that executes each subgoal.

Weighted value learning. To obtain this subgoal-ranking signal, we use w_S to train the value function under the corrected state occupancy. Following HIQL (Park et al., 2023), we learn a state value

V_ψ with an action-free expectile backup, so that the value compares candidate future states without assigning scores to individual logged actions. We can also learn V_ψ using OTA-style backups (Ahn et al., 2025), which bootstrap from future states through temporally extended trajectory segments, with rewards and discounting defined at the option level, as detailed in Appendix K.

We incorporate the occupancy correction by weighting each backup with $w_S(s;g)$, which changes the effective sampling distribution from the empirical state occupancy d_S^O to the corrected occupancy d_S^*:

$$\mathcal{L}_V(\psi) = \mathbb{E}_{(s,s',g)\sim\mathcal{D}_O}\Big[\mathrm{sg}\big(w_S(s;g)\big)\cdot L_2^\tau\big(r(s;g)+\gamma\bar{V}_\psi(s';g)-V_\psi(s;g)\big)\Big]. \tag{12}$$

With exact w_S, this loss reweights value regression under d_S^* while preserving the conditional expectile target. Proposition L.5 characterizes this weighted regression for a fixed target. Because (12) depends only on state transitions, the learned value can then be used directly to evaluate future states as high-level subgoals. This value function connects the occupancy correction to the hierarchical actor. It defines the high-level advantage $A_{\mathrm{hi}}(s_t, s_{t+k}; g)$ used to prefer subgoals that make progress toward g, while w_{SA} corrects the low-level action model on state-action pairs. The resulting actor first samples a subgoal from the high-level flow $\pi^{\mathrm{hi}}(z \mid s_t, g)$ and then generates primitive actions with the low-level flow $\pi^{\mathrm{lo}}(a_t \mid s_t, s_{t+k})$.

Both levels use the flow-matching template from Section 2 (Equation (3)); they differ only in the endpoint, conditioning variables, and importance weight.

High-level flow. At the high level, the flow endpoint is the future state $x^1 = s_{t+k}$, conditioned on the current state and task goal (s_t, g). The training objective upweights future states whose corrected value improves over the current state, using an exponential advantage weight:

$$\mathcal{L}_{\mathrm{hi}}(\theta_{\mathrm{hi}}) = \mathbb{E}_{(s_t,s_{t+k},g)}\Big[w_{\mathrm{hi}}\mathbb{E}_{\sigma,x^0}\big[\|v_{\theta_{\mathrm{hi}}}(\sigma,s_t,x^\sigma;g)-(x^1-x^0)\|^2\big]\Big], \tag{13}$$

$$w_{\mathrm{hi}} = \exp\big(\beta_{\mathrm{hi}}A_{\mathrm{hi}}(s_t,s_{t+k};g)\big), \quad A_{\mathrm{hi}}(s_t,s_{t+k};g) = \bar{V}_\psi(s_{t+k};g)-\bar{V}_\psi(s_t;g), \tag{14}$$

where β_{hi} is the inverse-temperature hyperparameter. Because the advantage is computed from the w_S-corrected value, w_S influences the high-level loss through V_ψ rather than through an additional outer weight. This placement preserves the role of the high-level actor: it should propose reachable trajectory futures, then select among them using the corrected value. We weight value regression by $w_S(s_t;g)$ and use the learned value function to compute advantage weights for training the high-level actor.

Low-level flow. At the low level, the flow endpoint is the logged action $x^1 = a_t$, conditioned on the current state and selected subgoal (s_t, s_{t+k}). Here the relevant correction is state-action based, since the model must decide which logged actions are useful for reaching the chosen subgoal. We therefore evaluate the ratio as $w_{SA}(s_t, a_t; s_{t+k})$, using the subgoal as the conditioning goal.

$$\mathcal{L}_{\mathrm{lo}}(\theta_{\mathrm{lo}}) = \mathbb{E}_{(s_t,a_t,s_{t+k})}\Big[w_{SA}(s_t,a_t;s_{t+k})\mathbb{E}_{\sigma,x^0}\big[\|v_{\theta_{\mathrm{lo}}}(\sigma,s_t,x^\sigma;s_{t+k})-(x^1-x^0)\|^2\big]\Big]. \tag{15}$$

One-step policy extraction. Following FQL (Park et al., 2025c), we distill the occupancy-weighted low-level flow into a one-step student that maps Gaussian noise directly to an action conditioned on the current state and subgoal. A critic learned from the w_S-corrected value provides additional Q-guidance during student training. This enables action generation in a single forward pass; the training objectives are provided in Appendix D.1.

4 Experiments

To evaluate the effectiveness of our dual-level occupancy weighting framework, we organize our main experiments around three questions: (Q1) How well does D^{2}FM compare with existing offline GCRL methods across diverse tasks? (Q2) How effectively does D^{2}FM handle tasks that require stitching? (Q3) How well does D^{2}FM learn from noisy offline datasets?

Benchmark. We evaluate D^{2}FM on six state-based environments from OGBench (Park et al., 2025a): `pointmaze` and `antmaze` for navigation, `antsoccer` for locomotion with object interaction, `cube` and `scene` for robotic manipulation, and `puzzle` for combinatorial planning. Our maze

Table 2: **Goal-conditioned success rates on OGBench.** We report each method's average (binary) success rate (%) across five test-time goals. We report D^{2}FM over five random seeds, with standard deviations after the $\pm$ sign. Numbers at or above 95% of the best performance per task are boldfaced, as in OGBench.

Task	Flat								Hierarchical				
	QRL	CRL	GCIQL	GCBC	GCIVL	CGCIVL	CFGRL	GCFBC	HIQL	OTA	HCFGRL	HGCFBC	D^{2}FM
Locomotion													
pointmaze													
medium-navigate	82 ±5	29 ±7	53 ±8	9 ±6	63 ±6	87 ±4	77 ±6	66 ±4	79 ±5	86 ±2	63 ±5	57 ±7	**100** ±0
large-navigate	86 ±9	39 ±7	34 ±3	29 ±6	45 ±5	58 ±15	77 ±5	74 ±7	58 ±5	85 ±5	57 ±8	75 ±6	**100** ±0
giant-navigate	68 ±7	27 ±10	0 ±0	1 ±2	0 ±0	80 ±12	30 ±10	4 ±2	46 ±9	72 ±6	18 ±8	6 ±5	**91** ±1
antmaze													
medium-navigate	88 ±3	**95** ±1	71 ±4	29 ±4	72 ±8	**95** ±1	53 ±12	42 ±9	**96** ±1	**96** ±1	90 ±5	67 ±8	**96** ±1
large-navigate	75 ±6	83 ±4	34 ±4	24 ±2	16 ±5	**91** ±2	24 ±10	22 ±5	**91** ±2	**92** ±1	78 ±4	61 ±5	**90** ±3
giant-navigate	14 ±3	16 ±3	0 ±0	0 ±0	0 ±0	73 ±5	1 ±1	0 ±0	65 ±5	**77** ±4	38 ±7	14 ±5	**75** ±5
antsoccer													
arena-navigate	8 ±2	23 ±2	50 ±2	5 ±1	47 ±3	60 ±2	15 ±6	10 ±5	58 ±2	38 ±3	37 ±5	16 ±5	**65** ±3
Locomotion avg	60.1	44.6	34.6	13.9	34.7	77.7	39.6	31.1	70.4	78.0	54.4	42.3	**88.1**
Manipulation													
cube													
double-play	1 ±0	10 ±2	40 ±5	1 ±1	36 ±3	**46** ±4	2 ±2	3 ±1	6 ±2	1 ±0	42 ±5	21 ±7	40 ±9
triple-play	1 ±0	4 ±1	3 ±1	1 ±1	1 ±0	5 ±1	7 ±5	3 ±1	6 ±2	1±0	1 ±0	3 ±2	**18** ±6
quadruple-play	0 ±0	0 ±0	0 ±0	0 ±0	0 ±0	0 ±0	0 ±0	0 ±0	0 ±0	0±0	0 ±0	0 ±0	**1** ±0
scene													
play	5 ±1	19 ±2	51 ±4	5 ±1	42 ±4	**77** ±5	19 ±4	15 ±5	38 ±3	15 ±2	18 ±5	14 ±4	73 ±8
puzzle													
4x4-play	0 ±0	0 ±0	26 ±3	0 ±0	13 ±2	32 ±5	0 ±0	0 ±0	7 ±2	1±0	0 ±0	0 ±0	**47** ±3
4x5-play	0 ±0	1 ±0	**14** ±3	0 ±0	7 ±1	**15** ±3	2 ±1	1 ±0	7 ±2	1 ±1	0 ±0	0 ±0	11 ±1
4x6-play	0 ±0	4 ±1	**12** ±1	0 ±0	10 ±2	9 ±2	0 ±0	0 ±0	7 ±2	1 ±0	0 ±0	0 ±0	9 ±2
Manipulation avg	1.0	5.4	20.9	1.0	15.6	26.3	4.3	3.1	10.1	2.9	8.7	5.4	**28.4**
Overall avg	30.6	25.0	27.7	7.4	25.1	52.0	21.9	17.1	40.3	40.4	31.6	23.9	**58.3**

Table 3: **Success rates on OGBench stitching tasks.** Evaluation and reporting conventions follow Table 2.

Task	Flat							Hierarchical					
	QRL	GCIQL	GCBC	GCIVL	CRL	CFGRL	GCFBC	HIQL	HFIQL	OTA	HCFGRL	HGCFBC	D^{2}FM
pointmaze													
medium-stitch	80 ±12	21 ±9	23 ±18	70 ±14	0 ±1	37 ±3	47 ±10	74 ±6	65 ±5	75 ±5	20 ±8	48 ±9	**95** ±8
teleport-stitch	9 ±5	25 ±3	31 ±9	44 ±2	4 ±3	23 ±4	39 ±6	34 ±4	39 ±2	19±2	4 ±2	22 ±7	**49** ±1
antmaze													
medium-stitch	59 ±7	29 ±6	45 ±11	44 ±6	53 ±6	49 ±5	61 ±5	**94** ±1	54±2	**93** ±1	5 ±4	4 ±0	**91** ±2
teleport-stitch	24 ±5	17 ±2	31 ±6	**39** ±3	31 ±4	28 ±4	**39** ±5	36 ±2	36 ±2	**39**±1	4 ±0	3 ±0	**39** ±3
antsoccer													
arena-stitch	1 ±1	2 ±0	24 ±8	21 ±3	1 ±0	30 ±2	**47** ±6	15 ±1	42 ±1	22±2	1 ±0	1 ±0	**45** ±1
medium-stitch	0 ±0	0 ±0	2 ±1	1 ±0	0 ±0	5 ±1	**9** ±2	4 ±1	6 ±2	6±2	2 ±2	3 ±2	**8** ±2
Overall avg	28.8	15.7	26.0	36.5	14.8	28.7	40.3	42.8	40.3	42.3	6.0	13.5	**54.5**

experiments cover `medium`, `large`, and `giant` layouts, spanning different spatial scales and goal-reaching horizons. We also evaluate dataset variants with distinct behavioral characteristics. Alongside the `navigate` and `play` datasets, we consider `stitch` datasets that require composing behaviors across trajectories and `noisy` datasets collected with action noise, which provide broader state coverage and more variable behavior. These settings allow us to assess performance across task scales and data characteristics.

Baselines. We compare against four families of offline GCRL methods: (i) value-based methods (`GCIQL` (Kostrikov et al., 2022), `GCIVL` (Xu et al., 2022)) and `CGCIVL` (Ke et al., 2025); (ii) value-free methods (`CRL` (Eysenbach et al., 2022) and `QRL` (Wang et al., 2023a)); (iii) behavior-cloning methods, including `GCBC`,`GCFBC`, and `CFGRL` (Frans et al., 2025); and (iv) hierarchical methods, namely `HIQL` (Park et al., 2023), `OTA` (Ahn et al., 2025), `HGCFBC`, and `HCFGRL` (Frans et al., 2025). On noisy and stitching tasks, we additionally evaluate `HFIQL`, an ablated variant of our method that uses IQL-style advantage weighting, to examine the performance of hierarchical flow policies in settings requiring trajectory stitching. Where available, baseline numbers are taken directly from (Park et al., 2025a). For methods not reported there, we run the authors' official implementations using the best-performing hyperparameter configurations reported by the authors and follow the same evaluation protocol. Additional details on the environments, datasets, and baselines are provided in Appendix E.

Table 4: **Success rates on OGBench tasks with noisy offline data.** Evaluation and reporting conventions follow Table 2.

Task	Flat							Hierarchical					
	QRL	GCIQL	GCBC	GCIVL	CRL	CFGRL	GCFBC	HIQL	HFIQL	OTA	HCFGRL	HGCFBC	D^{2}FM
cube													
double-noisy	3 ±1	23 ±3	1 ±1	14 ±3	2 ±1	7 ±1	7 ±1	2 ±1	22 ±1	4±1	6 ±3	5 ±0	**27** ±2
triple-noisy	1 ±0	**9** ±1	1 ±1	1 ±0	4 ±1	4 ±1	2 ±0	2 ±1	4 ±1	1±0	1 ±0	3 ±1	6 ±1
scene													
noisy	9 ±2	26 ±2	1 ±1	26 ±2	1 ±1	17 ±4	14 ±1	25 ±4	53 ±2	25±1	41 ±0	16 ±2	**60** ±6
puzzle													
4x4-noisy	0 ±0	29 ±7	0 ±0	20 ±3	0 ±0	1 ±0	0 ±0	16 ±4	20 ±1	12±2	0 ±0	0 ±0	**35** ±2
Manipulation avg	3.3	21.8	0.8	15.3	1.8	7.3	5.8	11.3	24.8	10.5	12.0	6.0	**32.0**

4.1 Main Results

Tables 2, 3, and 4 report success rates on 14 navigation, manipulation, and planning tasks, six stitching tasks, and four noisy-data tasks, respectively. Within each table, we group methods by policy structure into flat and hierarchical approaches and organize tasks by environment, enabling comparisons across policy structures and task settings.

Answer to Q1: D^{2}FM achieves the strongest average success rate across task categories. Table 2 shows that D^{2}FM achieves the highest average performance on both locomotion and manipulation tasks. On locomotion tasks, it achieves 88.1%, compared with 77.9% for the strongest baseline, OTA. On manipulation tasks, it achieves 28.4%, surpassing the strongest baseline, CGCIVL, at 26.3%. On `antmaze-{medium,large}-navigate`, D^{2}FM performs comparably to CGCIVL, HIQL, and OTA. We hypothesize that the high success rates already achieved by these baselines leave limited headroom for improvement, so occupancy correction may provide only modest additional benefits on these tasks. Overall, these results demonstrate the effectiveness of D^{2}FM across diverse goal-reaching tasks spanning locomotion, manipulation, and combinatorial planning.

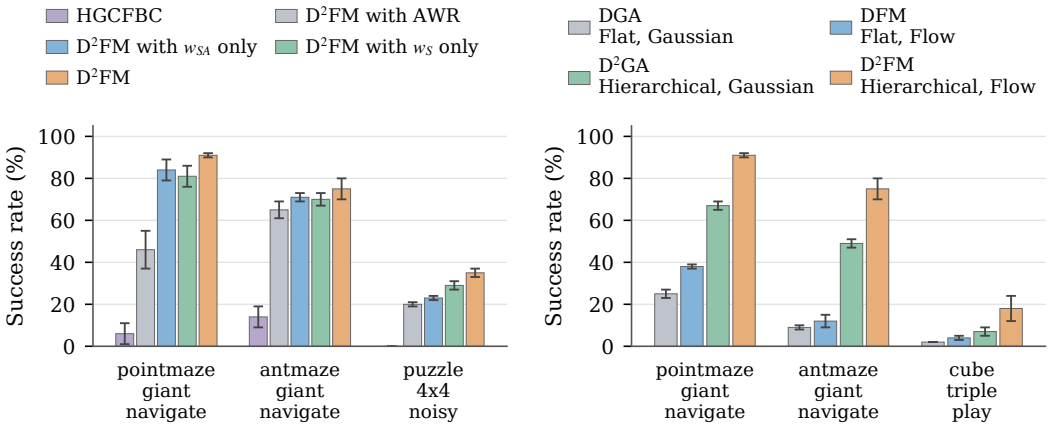
Answer to Q2: D^{2}FM achieves strong stitching performance with occupancy-weighted flow policies. The `stitch` datasets require composing partial behaviors across trajectories to reach the evaluation goals. Table 3 shows that D^{2}FM achieves the highest average success rate of 54.5%, compared with 42.8% for the strongest baseline, HIQL. We observe that hierarchy improves average performance with Gaussian policies (HIQL versus GCIVL), while flow parameterization benefits flat policies (e.g., GCFBC versus GCBC). However, combining hierarchy with flow policies does not consistently improve performance: HFIQL underperforms HIQL on average, and HCFGRL and HGCFBC trail their flat counterparts. This pattern suggests that high-level subgoal selection can limit the benefits of expressive policies when proposed subgoals provide insufficient progress toward the final goal. With state-weighted value learning for subgoal selection and state–action-weighted flow matching for low-level policy extraction, D^{2}FM consistently outperforms HFIQL across all six tasks.

Answer to Q3: D^{2}FM effectively learns goal-reaching policies from noisy offline data. Table 4 evaluates learning from trajectories collected with action noise, which provide broader state coverage and more variable behavior (Park et al., 2025a). These datasets challenge policy extraction, as the learner must identify behavior useful for reaching a given goal. D^{2}FM outperforms HFIQL on all four tasks, achieving an average success rate of 32.0% compared with 24.8%. The low success rates of GCBC and GCFBC suggest that unweighted imitation is insufficient to exploit this behavioral diversity. Flow-based HFIQL improves over Gaussian HIQL, indicating the benefit of expressive policies. The further gains of D^{2}FM are consistent with the role of dual occupancy weighting: emphasizing goal-relevant state–action samples during policy extraction and reweighting value learning to guide high-level subgoal selection.

4.2 Ablation Study

Our ablations examine two questions: (Q1) How do the two occupancy weights affect performance, separately and together? (Q2) How do policy architecture and policy class affect performance within our occupancy-weighted framework? For the first question, Figure 2(a) compares HGCFBC, D^{2}FM with AWR, D^{2}FM with w_{SA} only, D^{2}FM with w_S only, and the full D^{2}FM on PointMaze, AntMaze and Puzzle tasks, keeping the hierarchical flow architecture fixed. For the second question, Figure 2(b)

compares flat and hierarchical policies with Gaussian or flow parameterizations. Table 8 summarizes these configurations and their occupancy-weighting schemes.



(a) Dual occupancy weighting (b) Policy architecture and policy class

Figure 2: **Ablation studies. (a)** Dual occupancy weighting under a fixed hierarchical flow architecture. **(b)** Flat and hierarchical Gaussian or flow policies using DICE weights. The configurations in (b) are summarized in Table 8 in Appendix E.3.

Answer to Q1: Dual occupancy weighting improves performance under a fixed hierarchical flow architecture. Figure 2(a) compares five variants on three navigation and noisy puzzle tasks. The full D^{2}FM achieves 67.0% average success, compared with 59.3% for w_{SA} only, 60.0% for w_S only, 43.7% with AWR, and 6.7% for HGCFBC. It achieves the highest mean success rate on all three tasks. On the noisy puzzle task, w_S only outperforms w_{SA} only, suggesting that value correction for subgoal selection is particularly important in this setting. These comparisons support the complementary roles of occupancy correction in policy extraction and value learning.

Answer to Q2: Hierarchical flow policies achieve the strongest performance within our framework.

Figure 2(b) compares policy architectures and parameterizations on `pointmaze-giant-navigate`, `antmaze-giant-navigate`, and `cube-triple-play`. Replacing Gaussian policies with flow policies increases the average success rate from 12.0% to 18.0% in the flat setting and from 41.0% to 61.3% in the hierarchical setting. Hierarchical variants outperform their flat counterparts on all three tasks, with D^{2}FM achieving the highest mean success rate on each. The larger benefit of flow parameterization in the hierarchical setting supports combining expressive flow policies with subgoal-based decomposition within our occupancy-weighted framework.

Additional experiments. (i) **Runtime and memory usage.** Training time and peak GPU memory in Appendix I show that D^{2}FM incurs only modest overhead hierarchical flow baselines. (ii) **Hyperparameter sensitivity.** Sweeps over β_{hi}, α, and λ_q in Appendix J show D^{2}FM is stable across a wide range. (iii) **Subgoal step horizon.** The horizon sweep in Appendix G empirically shows that harder longer-horizon tasks may prefer larger subgoal steps k.

5 Conclusion

We presented *Doubly Dual-Weighted Flow Matching* (D^{2}FM), a hierarchical offline GCRL method that derives two coupled occupancy ratios from a shared reward-tilted target. The state–action ratio $w^\star_{SA}$ weights low-level conditional flow matching, while the state ratio $w^\star_S$ weights action-free value learning. Experiments on OGBench demonstrate strong performance across diverse tasks, and ablations support the complementary benefits of the two correction ratios.

AI USE STATEMENT

Generative-AI tools assisted with literature organization, drafting, and consistency checks. The authors manually verified every cited source and mathematical argument, and reviewed all AI-assisted artifacts. The authors take responsibility for the final content.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we provide the hyperparameter settings, implementation details, and algorithm pseudocode in the appendix, documenting the training procedure and experimental configurations used to obtain the reported results.

REFERENCES

Hongjoon Ahn, Heewoong Choi, Jisu Han, and Taesup Moon. Option-aware temporally abstracted value for offline goal-conditioned reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, et al. Jax: Autograd and xla. *Astrophysics Source Code Library*, 2021.

Elliot Chane-Sane, Cordelia Schmid, and Ivan Laptev. Goal-conditioned reinforcement learning with imagined subgoals. In *International conference on machine learning*, pp. 1430–1440. PMLR, 2021.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems*, 2023.

Jinwoo Choi, Sang-Hyun Lee, and Seung-Woo Seo. Chain-of-goals hierarchical policy for long-horizon offline goal-conditioned rl. *arXiv preprint arXiv:2602.03389*, 2026.

Benjamin Eysenbach, Tianjun Zhang, Sergey Levine, and Ruslan Salakhutdinov. Contrastive learning as goal-conditioned reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Kevin Frans, Seohong Park, Pieter Abbeel, and Sergey Levine. Diffusion guidance is a controllable policy improvement operator. *arXiv preprint arXiv:2505.23458*, 2025.

Dibya Ghosh, Abhishek Gupta, Ashwin Reddy, Justin Fu, Coline Devin, Benjamin Eysenbach, and Sergey Levine. Learning to reach goals via iterated supervised learning. In *International Conference on Learning Representations*, 2021.

Dibya Ghosh, Chethan Anand Bhateja, and Sergey Levine. Reinforcement learning from passive data via latent intentions. In *International Conference on Machine Learning*, pp. 11321–11339. PMLR, 2023.

Abhishek Gupta, Vikash Kumar, Corey Lynch, Sergey Levine, and Karol Hausman. Relay policy learning: Solving long horizon tasks via imitation and reinforcement learning. *Conference on Robot Learning (CoRL)*, 2019.

Dan Haramati, Carl Qi, Tal Daniel, Amy Zhang, Aviv Tamar, and George Konidaris. Hierarchical entity-centric reinforcement learning with factored subgoal diffusion. *arXiv preprint arXiv:2602.02722*, 2026.

Joey Hejna, Jensen Gao, and Dorsa Sadigh. Distance weighted supervised learning for offline interaction data. In *International Conference on Machine Learning*, pp. 12882–12906. PMLR, 2023.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

Christopher Hoang, Sungryull Sohn, Jongwook Choi, Wilka Carvalho, and Honglak Lee. Successor feature landmarks for long-horizon goal-conditioned reinforcement learning. *Advances in neural information processing systems*, 34:26963–26975, 2021.

Abhinav Jain and Vaibhav Unhelkar. Go-dice: Goal-conditioned option-aware offline imitation learning via stationary distribution correction estimation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 12763–12772, 2024.

Michael Janner, Yilun Du, Joshua Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *International Conference on Machine Learning*, 2022.

Kaiqiang Ke, Qian Lin, Zongkai Liu, Shenghong He, and Chao Yu. Conservative offline goal-conditioned implicit v-learning. In *ICML*, 2025.

Sungyoon Kim, Yunseon Choi, Daiki E Matsunaga, and Kee-Eung Kim. Stitching sub-trajectories with conditional diffusion model for goal-conditioned offline rl. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 13160–13167, 2024.

Woosung Kim, JunHo Seo, Jongmin Lee, and Byung-Jun Lee. Semi-gradient dice for offline constrained reinforcement learning. *arXiv preprint arXiv:2506.08644*, 2025.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit Q-learning. In *International Conference on Learning Representations*, 2022.

Aviral Kumar, Joey Hong, Anikait Singh, and Sergey Levine. When should we prefer offline reinforcement learning over behavioral cloning? *arXiv preprint arXiv:2204.05618*, 2022.

Jongmin Lee, Wonseok Jeon, Byung-Jun Lee, Joelle Pineau, and Kee-Eung Kim. Optidice: Offline policy optimization via stationary distribution correction estimation. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021.

Lisa Lee, Benjamin Eysenbach, Emilio Parisotto, Eric Xing, Sergey Levine, and Ruslan Salakhutdinov. Efficient exploration via state marginal matching. *arXiv preprint arXiv:1906.05274*, 2019.

Jinning Li, Chen Tang, Masayoshi Tomizuka, and Wei Zhan. Hierarchical planning through goal-conditioned offline reinforcement learning. *IEEE Robotics and Automation Letters*, 7(4):10216–10223, 2022.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *International Conference on Learning Representations*, 2023.

Yecheng Jason Ma, Andrew Shen, Dinesh Jayaraman, and Osbert Bastani. Versatile offline imitation from observations and examples via regularized state-occupancy matching. In *International Conference on Machine Learning*, pp. 14639–14663. PMLR, 2022a.

Yecheng Jason Ma, Jason Yan, Dinesh Jayaraman, and Osbert Bastani. How far i'll go: Offline goal-conditioned reinforcement learning via f-advantage regression. *arXiv preprint arXiv:2206.03023*, 2022b.

Ofir Nachum and Bo Dai. Reinforcement learning via fenchel-rockafellar duality. *arXiv preprint arXiv:2001.01866*, 2020.

Ofir Nachum, Yinlam Chow, Bo Dai, and Lihong Li. DualDICE: Behavior-agnostic estimation of discounted stationary distribution corrections. In *Advances in Neural Information Processing Systems*, 2019a.

Ofir Nachum, Bo Dai, Ilya Kostrikov, Yinlam Chow, Lihong Li, and Dale Schuurmans. AlgaeDICE: Policy gradient from arbitrary experience. In *arXiv preprint arXiv:1912.02074*, 2019b.

Ashvin Nair, Abhishek Gupta, Murtaza Dalal, and Sergey Levine. Awac: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020.

Seohong Park, Dibya Ghosh, Benjamin Eysenbach, and Sergey Levine. HIQL: Offline goal-conditioned RL with latent states as actions. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. Ogbench: Benchmarking offline goal-conditioned rl. In *International Conference on Learning Representations (ICLR)*, 2025a.

Seohong Park, Kevin Frans, Deepinder Mann, Benjamin Eysenbach, Aviral Kumar, and Sergey Levine. Horizon reduction makes rl scalable. *arXiv preprint arXiv:2506.04168*, 2025b.

Seohong Park, Qiyang Li, and Sergey Levine. Flow q-learning. In *International Conference on Machine Learning (ICML)*, 2025c.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. 2019.

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

Alfréd Rényi. On measures of entropy and information. In *Proceedings of the fourth Berkeley symposium on mathematical statistics and probability, volume 1: contributions to the theory of statistics*, pp. 547–562. University of California Press, 1961.

Quentin Rouxel, Clemente Donoso, Fei Chen, Serena Ivaldi, and Jean-Baptiste Mouret. Extremum flow matching for offline goal conditioned reinforcement learning. In *2025 IEEE-RAS 24th International Conference on Humanoid Robots (Humanoids)*, pp. 981–988. IEEE, 2025.

Harshit Sikchi, Rohan Chitnis, Ahmed Touati, Alborz Geramifard, Amy Zhang, and Scott Niekum. Smore: Score models for offline goal-conditioned reinforcement learning. *arXiv preprint arXiv:2311.02013*, 2023.

Jost Tobias Springenberg, Abbas Abdolmaleki, Jingwei Zhang, Oliver Groth, Michael Bloesch, Thomas Lampe, Philemon Brakel, Sarah Bechtle, Steven Kapturowski, Roland Hafner, et al. Offline actor-critic reinforcement learning scales to large models. *arXiv preprint arXiv:2402.05546*, 2024.

Masatoshi Uehara, Chengchun Shi, and Nathan Kallus. A review of off-policy evaluation in reinforcement learning. *arXiv preprint arXiv:2212.06355*, 2022.

Tongzhou Wang, Antonio Torralba, Phillip Isola, and Amy Zhang. Optimal goal-reaching reinforcement learning via quasimetric learning. In *International Conference on Machine Learning*, pp. 36411–36430. PMLR, 2023a.

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*, 2023b.

Haoran Xu, Li Jiang, Jianxiong Li, and Xianyuan Zhan. A policy-guided imitation approach for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, 2022.

Mengjiao Yang, Dale Schuurmans, Pieter Abbeel, and Ofir Nachum. Dichotomy of control: Separating what you can control from what you cannot. *arXiv preprint arXiv:2210.13435*, 2022a.

Rui Yang, Yiming Lu, Wenzhe Li, Hao Sun, Meng Fang, Yali Du, Xiu Li, Lei Han, and Chongjie Zhang. Rethinking goal-conditioned supervised learning and its connection to offline RL. In *International Conference on Learning Representations*, 2022b.

Rui Yang, Yong Lin, Xiaoteng Ma, Hao Hu, Chongjie Zhang, and Tong Zhang. What is essential for unseen goal generalization of offline goal-conditioned RL? In *International Conference on Machine Learning*, 2023.

Shiyuan Zhang, Weitong Zhang, and Quanquan Gu. Energy-weighted flow matching for offline reinforcement learning. In *International conference on learning representations*, volume 2025, pp. 17943–17970, 2025.

Tianjun Zhang, Benjamin Eysenbach, Ruslan Salakhutdinov, Sergey Levine, and Joseph E Gonzalez. C-planning: An automatic curriculum for learning goal-reaching tasks. *arXiv preprint arXiv:2110.12080*, 2021.

John L Zhou and Jonathan C Kao. Flattening hierarchies with policy bootstrapping. *arXiv preprint arXiv:2505.14975*, 2025.

Hanlin Zhu and Amy Zhang. Provably efficient offline goal-conditioned reinforcement learning with general function approximation and single-policy concentrability. *Advances in Neural Information Processing Systems*, 36:4177–4198, 2023.

A Detailed Preliminaries

Goal-conditioned MDP. We consider an infinite-horizon MDP (Puterman, 2014) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, R, T, \mu_0, \gamma)$ with state space $\mathcal{S}$, action space $\mathcal{A}$, reward function R, transition kernel $T(s' \mid s, a)$, initial distribution $\mu_0(s)$, and discount factor $\gamma \in (0, 1)$. In goal-conditioned RL, the MDP additionally specifies a goal space $G \coloneqq \{\phi(s) \mid s \in \mathcal{S}\}$ with a known mapping $\phi : \mathcal{S} \to G$. The reward $r(s; g)$[2] and the policy $\pi(a \mid s, g)$ are both conditioned on the goal $g \in G$. Given a goal distribution $p(g)$, the objective is to maximize the expected discounted return:

$$J(\pi) \coloneqq \mathbb{E}_{g\sim p(g), s_0\sim\mu_0, a_t\sim\pi(\cdot|s_t,g), s_{t+1}\sim T(\cdot|s_t,a_t)}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t; g)\right]. \tag{16}$$

The goal-conditioned state-action occupancy $d^\pi_{SA}(s, a; g) : \mathcal{S} \times \mathcal{A} \times G \to [0, 1]$ induced by π is

$$d^\pi_{SA}(s, a; g) \coloneqq (1-\gamma)\sum_{t=0}^{\infty}\gamma^t \Pr\big(s_t = s,\ a_t = a \,\big|\, \pi, T, \mu_0, g\big), \tag{17}$$

with state marginal $d^\pi_S(s; g) = \sum_a d^\pi_{SA}(s, a; g)$ and the identity $\pi(a \mid s, g) = d^\pi_{SA}(s, a; g)/d^\pi_S(s; g)$. Any valid state-action occupancy must satisfy the *Bellman flow constraint*:

$$\sum_a d_{SA}(s, a; g) = (1-\gamma)\mu_0(s) + \gamma\sum_{\tilde{s},\tilde{a}} T(s \mid \tilde{s}, \tilde{a})\, d_{SA}(\tilde{s}, \tilde{a}; g), \qquad \forall s \in \mathcal{S},\ g \in G. \tag{18}$$

Writing $d^\pi(s, g) = p(g)\, d^\pi_S(s; g)$ for the joint goal-state density induced by $p(g)$ and π, the objective in equation 1 can be rewritten as $J(\pi) = \frac{1}{1-\gamma}\mathbb{E}_{(s,g)\sim d^\pi(s,g)}[r(s; g)]$. In *offline GCRL*, the agent cannot interact with the environment and must learn from a static dataset $\mathcal{D}_O \coloneqq \{\tau^{(i)}\}_{i=1}^N$ of logged trajectories $\tau^{(i)} = (s^{(i)}_0, a^{(i)}_0, r^{(i)}_0, s^{(i)}_1, \ldots; g^{(i)})$, with $s^{(i)}_0 \sim \mu_0$. Trajectories need not come from a *goal-directed* policy; the goal $g^{(i)}$ may be drawn from $p(g)$ independently of the trajectory. We denote the empirical goal-conditioned state-action occupancy of $\mathcal{D}_O$ by $d^O(s, a; g)$, and estimate expectations $\mathbb{E}_{(s,a,g)\sim d^O}[\cdot]$ by uniform sampling from $\mathcal{D}_O$.

Convex conjugates and f-divergence We briefly introduce f-divergences and their Fenchel conjugates. Let $f : \mathbb{R}_+ \to \mathbb{R}$ be a convex function. The convex conjugate $f^* : \mathbb{R}_+ \to \mathbb{R}$ of f is defined by:

$$f^*(y) = \sup_{x\in\mathbb{R}_+}[xy - f(x)]. \tag{19}$$

The convex conjugates have the important property that f^* is also convex and the convex conjugate of f^* retrieves back the original function f, if f is proper, lower semicontinuous, and convex. We also note an important relation regarding f and f^*: $(f^*)' = (f')^{-1}$, where the $'$ notation denotes first derivative.

We next use f-divergences to compare probability distributions. Formally, an f-divergence (Rényi, 1961) is defined as follows:

Let P and Q be two probability distributions over a space $\mathcal{Z}$ such that P is absolutely continuous with respect to Q [3]. For a function $f : \mathbb{R}_+ \to \mathbb{R}$ that is convex, lower-semicontinuous and $f(1) = 0$, the f-divergence of P from Q is

$$D_f(P\|Q) = \mathbb{E}_{z\sim Q}\left[f\left(\frac{P(z)}{Q(z)}\right)\right]. \tag{20}$$

Table 5 lists some common f-divergences with their generator functions f and the conjugate functions f^*.

Hierarchical Implicit Q-Learning (HIQL). HIQL (Park et al., 2023) decomposes goal reaching into high-level subgoal selection and low-level action execution, using an action-free value function

[2] In GCRL the reward customarily does not depend on the action.

[3] Let z denote the random variable. For any measurable set $Z \subseteq \mathcal{Z}$, $Q(z \in Z) = 0$ implies $P(z \in Z) = 0$.

Divergence Name	Generator $f(x)$	Conjugate $f^*(y)$
Reverse KL	$x \log x$	$e^{(y-1)}$
Squared Hellinger	$(\sqrt{x}-1)^2$	$\frac{y}{1-y}$
Pearson χ^2	$(x-1)^2$	$y + \frac{y^2}{4}$
Total Variation	$\frac{1}{2}\lvert x-1\rvert$	y if $y \in [-\frac{1}{2}, \frac{1}{2}]$ otherwise ∞
Jensen-Shannon	$-(x+1)\log(\frac{x+1}{2}) + x \log x$	$-\log(2-e^y)$

Table 5: List of common f-divergences.

for policy extraction at both levels. The goal-conditioned value functions $V_\psi(s;g), Q_\phi(s,a;g)$ are trained with the expectile loss[4]

$$\mathcal{L}_V(\psi) = \mathbb{E}_{(s,s')\sim\mathcal{D}_O,\, g\sim p(g)}\big[L_2^\tau\big(r(s;g) + \gamma V_{\bar{\psi}}(s';g) - V_\psi(s;g)\big)\big], \tag{21}$$

$$\mathcal{L}_Q(\phi) = \mathbb{E}_{(s,a,s')\sim\mathcal{D}_O,\, g\sim p(g)}\left[\big(r(s;g) + \gamma V_\psi(s';g) - Q_\phi(s,a;g)\big)^2\right] \tag{22}$$

where $L_2^\tau(x) = |\tau - \mathbb{1}(x<0)|\, x^2$ is the expectile loss with parameter $\tau \in [0.5, 1)$, and $\bar{\psi}$ denotes the target-network parameters. Because equation 21 uses only state transitions, V_ψ can be learned from action-free data. HIQL then defines a high-level advantage $A_{\text{hi}}(s_t, s_{t+k}; g) = V_\psi(s_{t+k};g) - V_\psi(s_t;g)$ to rank future states as candidate subgoals s_{t+k}, and extracts both a high-level policy $\pi_{\theta_{hi}}(s_{t+k} \mid s, g)$ and a low-level policy $\pi_{\theta_{lo}}(a \mid s, s_{t+k})$ via advantage-weighted regression (AWR):

$$\mathcal{L}_\pi(\theta_{lo}) = \mathbb{E}_{(s,a,s')\sim\mathcal{D}_O,\, g\sim p(g)}\big[\exp\big(\beta_{\text{hi}} \cdot A_{\text{lo}}(s,a;g)\big)\, \log \pi_{\theta_{lo}}(a \mid s, g)\big] \tag{23}$$

where $A_{\text{lo}}(s,a;g) = Q_\phi(s,a;g) - V_\psi(s;g)$ and β_{hi} is the advantage weighting temperature.

Flow Matching Policy. Recent work (Park et al., 2025c; Frans et al., 2025) applies flow matching (Lipman et al., 2023) to policy learning. The behavior-cloning flow-matching objective is

$$\mathcal{L}_{\text{Flow}}(\theta) = \mathbb{E}_{x^1\sim\mathcal{D},\, x^0\sim\mathcal{N}(0,I_d),\, \sigma\sim\text{Unif}([0,1])}\big[\|v_\theta(\sigma, s, x^\sigma) - (x^1 - x^0)\|_2^2\big], \tag{24}$$

where $v_\theta(\sigma, s, x^\sigma)$ is a state- and time-dependent vector field. The induced flow $\Phi_\theta(\sigma, s, x)$ defines a state-conditioned policy: for $s \in \mathcal{S}$ and noise $z \in \mathbb{R}^d$, we denote the ODE output $\Phi_\theta(1, s, z)$ by $\mu_\theta(s, z)$, which maps noise z to an action $a = \mu_\theta(s, z)$.

HCFGRL (Frans et al., 2025) extends the flow-matching policy to a hierarchical setting with a high-level subgoal flow and a low-level action flow, both trained on unweighted offline data. To bias generation toward goal-reaching behavior, HCFGRL applies classifier-free guidance (CFG) at inference rather than reweighting at training time. During training, the conditioning input (the goal g at the high level, or the subgoal z at the low level) is dropped with probability p_{drop}, so that a single network jointly represents the conditional vector field $v_\theta(\sigma, x \mid c)$ and the unconditional one $v_\theta(\sigma, x \mid \emptyset)$. At inference, the two are combined into a guided vector field

$$\tilde{v}_\theta(\sigma, x \mid c) = (1+w)\, v_\theta(\sigma, x \mid c) - w\, v_\theta(\sigma, x \mid \emptyset), \tag{25}$$

where $w \geq 0$ is the guidance scale that amplifies the influence of the condition. Because guidance acts only at inference, the underlying flow is still fit to the unweighted behavior distribution, so any mismatch between the offline distribution and the optimal occupancy persists.

B Related Work

Weighting transitions in offline GCRL. One line of work in offline GCRL focuses on selecting and weighting logged transitions for goal-conditioned policy learning. HER (Andrychowicz et al.,

[4]This action-free value learning is unbiased under deterministic dynamics (Ghosh et al., 2021). Under stochastic dynamics it may overestimate values; we adopt the deterministic assumption following prior work (Ghosh et al., 2021; Xu et al., 2022; Ghosh et al., 2023). Handling stochastic environments via controllability decomposition (Yang et al., 2022a) is an interesting future direction.

2017) introduced hindsight goal relabeling, while GCSL (Ghosh et al., 2021) applies supervised imitation to actions conditioned on goals reached later in the same trajectory. Relabeling changes the goal-labeled tuples used for training, but does not itself assign learned importance weights to them. Explicit weighting appears in WGCSL (Yang et al., 2022b), which combines discounted relabeling and advantage-based weights. More broadly, AWR (Peng et al., 2019) and AWAC (Nair et al., 2020) provide advantage-weighted regression schemes; GCIQL also supports an AWR policy-extraction variant (Park et al., 2025a). Advantage-based weights can be affected by errors in the learned values, particularly with sparse rewards and distant goals. DICE-style methods instead derive occupancy-ratio weights from constrained distribution optimization, as discussed next. Our low-level flow uses such a dual-derived state–action ratio to weight relabeled transitions.

Occupancy matching and DICE. The DICE family (Nachum et al., 2019b;a; Lee et al., 2021) uses stationary-distribution correction and convex duality for offline policy evaluation and optimization. SMODICE (Ma et al., 2022a), GoFAR (Ma et al., 2022b), and SMORe (Sikchi et al., 2023) develop related occupancy-matching formulations for imitation learning and offline GCRL. GoFAR is the closest precursor to our formulation. Building on its dual-based occupancy optimization, we use a reward-tilted offline state distribution as the matching target, incorporating offline coverage into the target and making the density-ratio reward analytically available. Our hierarchical learner additionally recovers a marginal state ratio: the state–action ratio weights low-level conditional flow matching, while the state ratio reweights action-free value learning to guide high-level subgoal selection. CORSDICE (Kim et al., 2025) also recovers a state correction, but starts from a policy correction and uses it for cost evaluation in constrained offline RL. Our auxiliary objective instead takes a state–action occupancy correction as input. With this correction fixed, its linear contribution permits direct sample-based optimization, avoiding the nested conditional expectations inside the convex conjugate that motivate CORSDICE's bias-reduction procedure. GO-DICE (Jain & Unhelkar, 2024) combines goal conditioning and hierarchical distribution correction for offline imitation learning, learning option-transition and action-execution policies from demonstrations. Our method instead addresses reward-based offline GCRL through subgoal selection and flow-based action generation, with separate occupancy corrections for value learning and policy extraction.

Hierarchy for long-horizon offline GCRL. Hierarchical methods decompose a far goal into closer subgoals. HIQL (Park et al., 2023) is the canonical instance: it trains a high-level subgoal policy and a low-level action policy with AWR weights derived from an action-free value function. Its hierarchical flow extension HCFGRL (Frans et al., 2025) pairs hierarchy with a flow actor but uses unweighted behavior data and relies on classifier-free guidance at inference, which cannot correct a mis-specified prior. Recent work pursues alternative hierarchical inductive biases: OTA (Ahn et al., 2025) addresses order-inconsistency in hierarchical value functions; SAW (Zhou & Kao, 2025) flattens hierarchies via advantage-weighted bootstrapping from learned subpolicies; CoGHP (Choi et al., 2026) uses an autoregressive chain-of-goals formulation; HECRL (Haramati et al., 2026) combines subgoal diffusion with value-based selection. D^2FM retains the standard two-level decomposition, using a state–action occupancy ratio to reweight low-level flow matching and a marginal state ratio to reweight action-free value learning. The resulting value estimates guide high-level subgoal selection through advantage weighting.

Flow and diffusion policies for offline RL. Flow matching (Lipman et al., 2023) and diffusion (Wang et al., 2023b; Chi et al., 2023) parameterize expressive multimodal action distributions. Diffusion-QL (Wang et al., 2023b) regularizes a diffusion policy with a Q-function; QIPO (Zhang et al., 2025) uses Q-guided implicit optimization with a flow model; FQL (Park et al., 2025c) distills Q-values into a one-step flow policy; Extremum FM (Rouxel et al., 2025) applies flow matching with extremum-seeking objectives. We use conditional flow matching as an instantiation of the corrected hierarchical policy and adopt one-step distillation (Park et al., 2025c) to recover deployment efficiency.

C LIMITATIONS

As with other offline RL methods, D^2FM may be limited by sparse coverage of goal-relevant behaviors. In such settings, neural estimates of occupancy ratios and values may be less accurate, reducing the benefit of occupancy correction. Incorporating task structure or representations learned from broader datasets could help improve generalization across related states and goals.

D IMPLEMENTATION DETAILS

Flow matching. We adopt the standard linear-path conditional flow-matching objective with uniform time sampling. The low-level teacher flow is integrated with a 10-step Euler solver across all tasks, both to generate distillation targets during training (30) and as the multi-step baseline reported in Figure 3. At deployment, the low-level policy is the one-step distilled student described in §3.2; the high-level flow uses 5 steps at inference.

Goal distributions. The value function (21), high-level policy (13), and low-level policy (15) are trained jointly under different goal-sampling distributions. For the value function we follow (Ghosh et al., 2023) and draw the goal from a future state, a random state, or the current state with probabilities 0.5, 0.3, and 0.2 respectively. Future states are sampled from $\mathrm{Geom}(1-\gamma)$ along the same trajectory; random states are drawn uniformly from the offline dataset. For the hierarchical policies we adapt the sampling strategy of (Gupta et al., 2019). Given a trajectory $(s_0, \dots, s_T)$ from $\mathcal{D}_\mathcal{O}$ and a timestep t, the high-level policy (Equation (13)) uses one of two goal sources: (*i*) a future state s_{t_g} $(t_g > t)$ drawn uniformly from the same trajectory, with target subgoal $s_{\min(t+k,t_g)}$; or (*ii*) a random state drawn uniformly from the dataset, with target subgoal $s_{\min(t+k,T)}$. The low-level policy (Equation (15)) is trained on $(s_t, s_{\min(t+k,T)})$ pairs sampled from the same trajectory.

Advantage estimates. We adopt the simplified advantage estimates introduced in Park et al. (2023). The exact advantage estimates entering Equation (13) are

$$A^h(s_t, s_{t+\tilde{k}}, g) = \gamma^{\tilde{k}} V_\psi(s_{t+\tilde{k}}, g) + \sum_{t'=t}^{\tilde{k}-1} r(s_{t'}, g) - V_\psi(s_t, g), \tag{26}$$

$$A^\ell(s_t, a_t, \tilde{s}_{t+k}) = \gamma V_\psi(s_{t+1}, \tilde{s}_{t+k}) + r(s_t, \tilde{s}_{t+k}) - V_\psi(s_t, \tilde{s}_{t+k}), \tag{27}$$

where $\tilde{k}$ and $\tilde{s}_{t+k}$ absorb the edge cases of the previous paragraph: $\tilde{k} = \min(k, t_g - t)$ when g is a future state, $\tilde{k} = \min(k, T-t)$ when g is a random state, and $\tilde{s}_{t+k} = s_{\min(t+k,T)}$. Outside these edge cases, $s_{t'} \neq g$ and $s_t \neq \tilde{s}_{t+k}$, so under the sparse reward $r(s, g) = -\mathbb{I}[s \neq g]$ the reward sums in Equations (26) and (27) are constant with respect to the policy inputs, as are the bootstrapped value terms at s_t. We therefore drop both, and absorb the discount factors into the temperature β. The resulting estimates, which match the exact form to within evaluation noise in our runs, are

$$\tilde{A}^h(s_t, s_{t+\tilde{k}}, g) = V_\psi(s_{t+\tilde{k}}, g) - V_\psi(s_t, g), \tag{28}$$

$$\tilde{A}^\ell(s_t, a_t, \tilde{s}_{t+k}) = V_\psi(s_{t+1}, \tilde{s}_{t+k}) - V_\psi(s_t, \tilde{s}_{t+k}). \tag{29}$$

D.1 ONE-STEP POLICY EXTRACTION

The low-level teacher is trained by occupancy-weighted flow matching in equation 15. Generating an action from this teacher requires multiple velocity evaluations through Euler integration. Following FQL (Park et al., 2025c), we train a one-step student $h_\eta(s_t, z; s_{t+k})$ that maps Gaussian noise z directly to an action conditioned on the current state and subgoal.

Teacher–student distillation. Let $\mu_{\theta_{\mathrm{lo}}}(s_t, z; s_{t+k})$ denote the terminal action obtained by integrating the teacher flow from noise z. We use a 10-step Euler solver to construct these targets. The student matches the teacher's output under the same noise:

$$\mathcal{L}_{\mathrm{distill}}(\eta) = \mathbb{E}_{\substack{(s_t, s_{t+k}) \sim \mathcal{D}_O \\ z \sim \mathcal{N}(0,I)}} \left[\left\| h_\eta(s_t, z; s_{t+k}) - \mathrm{sg}(\mu_{\theta_{\mathrm{lo}}}(s_t, z; s_{t+k})) \right\|^2 \right], \tag{30}$$

where sg denotes the stop-gradient operator. The teacher is updated through equation 15, while the distillation loss updates only the student.

Critic learning and Q-guidance. We learn an action critic Q_ϕ by bootstrapping from the value function V_ψ trained with the w_S-weighted expectile loss in equation 12:

$$\mathcal{L}_Q(\phi) = \mathbb{E}_{\substack{(s,a,s') \sim \mathcal{D}_O \\ g \sim p(g)}} \left[\left(Q_\phi(s, a; g) - \mathrm{sg}[r(s; g) + \gamma V_\psi(s'; g)] \right)^2 \right]. \tag{31}$$

The critic then guides the student toward actions with higher predicted value for the conditioning subgoal:

$$\mathcal{L}_{\text{Q-guidance}}(\eta) = -\mathbb{E}_{\substack{(s_t, s_{t+k})\sim\mathcal{D}_O \\ z\sim\mathcal{N}(0,I)}} \left[Q_\phi(s_t, h_\eta(s_t, z; s_{t+k}); s_{t+k}) \right]. \tag{32}$$

During this update, the critic parameters are held fixed, while gradients propagate through its action input to the student.

Combined objective and deployment. The complete low-level training objective is

$$\mathcal{L}_{\text{comp}}(\theta_{\text{lo}}, \eta) = \mathcal{L}_{\text{lo}}(\theta_{\text{lo}}) + \lambda_d \mathcal{L}_{\text{distill}}(\eta) + \lambda_q \mathcal{L}_{\text{Q-guidance}}(\eta),$$

where λ_d and λ_q control the distillation and Q-guidance terms, respectively. State–action correction enters through the teacher's weighted flow-matching objective, while state correction influences Q-guidance through the value function used to train the critic. At deployment, the student generates low-level actions in a single forward pass.

E Experimental Details

We implement D^{2}FM and many of the baselines in JAX (Bradbury et al., 2021) on top of OGBench's reference implementations (Park et al., 2025a). We run our experiments on an internal GPU cluster composed of A800 GPUs.

E.1 Environments, Tasks and Datasets

We evaluate on OGBench (Park et al., 2025a), a recently introduced benchmark for offline goal-conditioned RL that includes a diverse suite of locomotion, manipulation, and combinatorial-planning domains together with curated, suboptimal datasets. From OGBench we use six state-based environments, summarized in Table 6. We follow the same evaluation protocol as OGBench and adopt its unmodified observation, action, and reward conventions, including the sparse reward $r(s; g) = \mathbb{I}[s = g] - 1$. This ensures that our results are directly comparable with the baseline results reported in OGBench.

Domains. We choose the six environments to span the failure modes that motivate D^{2}FM:

- **pointmaze**. A point-mass agent navigates a maze. Trivial low-level dynamics, so success is dominated by the quality of subgoal selection across long horizons.
- **antmaze**. A quadrupedal Ant must reach a target (x, y) location. Combines a high-dimensional locomotion controller with long-horizon planning.
- **antsoccer**. The Ant must dribble a ball to a desired location, requiring coordinated agent–object control on top of locomotion.
- **cube**. A robot arm manipulates one to four cubes between configurations; the number of cubes scales the task horizon and the multimodality of valid solutions.
- **scene**. A single arm manipulates multiple heterogeneous objects, with up to 8 sequential subtasks per episode; this is the longest-horizon manipulation domain.
- **puzzle**. A button-pressing task on 4×4, 4×5, and 4×6 grids, requiring combinatorial generalization across configurations rather than just long-horizon control.

Dataset variants. OGBench provides several dataset types per environment that differ in coverage and quality. We use:

- **`navigate`** (locomotion): trajectories that aim for random goals; standard coverage of the maze.
- **`play`** (manipulation): trajectories that manipulate random objects to random configurations; high suboptimality and many compatible action modes per state.
- **`stitch`**: short-horizon segments that do not individually solve any test goal but, when composed across trajectories, can reach the test goals. Designed to isolate trajectory-stitching ability.

- **noisy**: trajectories with action-level noise injected, lowering data quality and increasing the multimodality of behavior at any given state.

Dataset sizes (number of trajectories and total transitions) and episode horizons follow the public OGBench release (Park et al., 2025a).

Table 6: **Environments used in our experiments.** $|\mathcal{A}|$ denotes the action dimensionality.

Domain	Description	$\lvert\mathcal{A}\rvert$	Dataset variants used
`pointmaze`	2D point-mass maze navigation	2	navigate, stitch
`antmaze`	Quadruped maze navigation	8	navigate, stitch
`antsoccer`	Quadruped + ball dribbling	8	navigate, stitch
`cube`	Arm + 1–4 cubes manipulation	5	play, noisy
`scene`	Arm + multi-object long-horizon	5	play, noisy
`puzzle`	Button-grid combinatorial planning	5	play, noisy

E.2 Hyperparameters

We adopt OGBench's reference network architectures and goal-sampling defaults (Ghosh et al., 2023; Park et al., 2025a; 2023). All MLP layers use layer normalization (Ba et al., 2016) and the GELU nonlinearity (Hendrycks & Gimpel, 2016); all networks are optimized with Adam (Kingma & Ba, 2014) and weight decay 10^{-4} on a single A800-80GB GPU. For fairness, every baseline uses the same architecture, and all hierarchical methods share the same subgoal horizon k, using the default value given by HIQL[5]. We present the hyperparameters used in our experiments in Table 7.

Table 7: Common hyperparameters for D^{2}FM

Hyperparameter	Value
Learning rate	3e-4
Batch size	1024
Optimizer	Adam (Kingma & Ba, 2014)
Weight decay	10^{-4}
Gradient clipping (max norm)	10.0
Nonlinearity / normalization	GELU + LayerNorm
Value MLP dimensions	$(512, 512, 512)$
Actor (flow) MLP dimensions	$(512, 512, 512)$
Time embedding dimension (flow)	128
Discount γ	0.99(default), 0.995(Antmaze-giant, HumanoidMaze)
Target-network EMA	0.005
f-divergence	χ^2
Data constraint Coeff. α	1.0
IQL expectile (high-level value)	0.7
Low-level distillation coeff. λ_d	1.0
High-level flow steps	5
Low-level flow steps	10

Beyond HIQL's defaults, only the high-level temperature β_{hi} and the Q-guidance coefficient λ_q are tuned; we fix the data-constraint coefficient at $\alpha = 1.0$. The per-domain grids are $\beta_{hi} \in \{1, 3, 5, 10\}$ and $\lambda_q \in \{0.01, 0.1, 1.0, 10, 100\}$. A sensitivity study on `antmaze-medium-navigate` (Figure 4, Table 13) additionally varies α and shows that D^{2}FM is robust to the hyperparameters.

[5] https://github.com/seohongpark/ogbench/blob/master/impls/hyperparameters.sh

E.3 POLICY-ARCHITECTURE ABLATION CONFIGURATIONS

Table 8 summarizes the four variants evaluated in Figure 2(b).

Table 8: **Policy-architecture configurations.** Variants compared in Figure 2(b).

Variant	Structure	Policy	w_{SA}	w_S
DGA	Flat	Gaussian	✓	–
DFM	Flat	Flow	✓	–
D^{2}GA	Hierarchical	Gaussian	✓	✓
D^{2}FM	Hierarchical	Flow	✓	✓

F TRAINING ALGORITHM

The D^{2}FM framework is outlined in Algorithm 1. Training alternates between estimating the dual-derived occupancy ratios, updating the weighted value and critic networks, fitting the hierarchical flow policies, and distilling the low-level flow into a one-step student. This procedure couples distribution correction with policy learning: the state ratio w_S shapes high-level subgoal learning through the action-free value function, while the state-action ratio w_{SA} weights low-level flow matching. Finally, the learned critic provides an additional guidance signal for distillation, enabling efficient deployment without discarding multimodal action coverage.

Algorithm 1 Doubly Dual-Weighted Flow Matching (D^{2}FM)

Require: Dataset $\mathcal{D}_O$, EMA rate τ, temperatures α, β_{hi}, distill coeff λ_{d}, Q coeff λ_{q}

Initialize $\nu_\xi, U_\zeta, V_\psi, Q_\phi, v_{\theta_{\text{lo}}}, h_\eta, v_{\theta_{\text{hi}}}$

Initialize all EMA targets: $\bar{\nu} \leftarrow \nu_\xi$, $\bar{V} \leftarrow V_\psi$, $\bar{Q} \leftarrow Q_\phi$, $\bar{v}_{\text{hi}} \leftarrow v_{\theta_{\text{hi}}}$

for step $= 1, \ldots, T$ **do**

  Sample batch from $\mathcal{D}_O$; pair each transition with goals from trajectory futures, random states, and commands

  // Dual weights computation

  $w_{\text{SA}} \leftarrow \max\Big(0,\ (f')^{-1}\Big(\frac{\mathcal{T}\bar{\nu}(s,a;g)-\bar{\nu}(s;g)}{\alpha}\Big)\Big)$

  $w_S \leftarrow \max\big(0,\ (f')^{-1}(U_\zeta(s;g))\big)$

  // Value updates

  Update ν_ξ via $\mathcal{L}_{\text{DICE}}$ (Eq. 7)

  Update U_ζ via $\mathcal{L}_{\text{OPE}}$ (Eq. 10) with $\text{sg}(w_{\text{SA}})$

  Update V_ψ via w_S-weighted expectile regression $\mathcal{L}_V$ (Eq. 12)

  Update Q_ϕ via Bellman regression $\mathcal{L}_Q$ (Eq. 31)

  // Policy updates

  Compute w_{h} via AWR (Eq. 14)

  Update $v_{\theta_{\text{hi}}}$ via $\mathcal{L}_{\text{hi}}$ (Eq. 13)

  Update $v_{\theta_{\text{lo}}}, h_\eta$ via

$$\mathcal{L}_{\text{lo}} + \lambda_{\text{d}}\mathcal{L}_{\text{distill}} + \lambda_{\text{q}}\mathcal{L}_{\text{Q-guidance}} \tag{33}$$

  (Eqs. 15, 30, 32)

  // EMA target updates

  $\bar{\nu} \leftarrow \tau\nu_\xi + (1-\tau)\bar{\nu}$; $\bar{V} \leftarrow \tau V_\psi + (1-\tau)\bar{V}$; $\bar{Q} \leftarrow \tau Q_\phi + (1-\tau)\bar{Q}$; $\bar{v}_{\text{hi}} \leftarrow \tau v_{\theta_{\text{hi}}} + (1-\tau)\bar{v}_{\text{hi}}$

end for

return High-level flow v_θ^{hi}, one-step actor h_η

G SUBGOAL HORIZON

Beyond the choice of weighting rule, the hierarchy also depends on the temporal abstraction used to sample candidate subgoals. We vary the high-level subgoal horizon k and keep all other settings

fixed. Table 9 reports the result. Small k values reduce the hierarchy to nearly local control, while very large k values can make subgoal supervision sparse or difficult to realize.

Table 9: **Effect of subgoal step horizon.** Binary goal-reaching success rate (%) for different high-level subgoal step horizons.

Task	Subgoal steps k	Success rate (%)
antmaze-large-navigate	1	1
antmaze-large-navigate	10	70
antmaze-large-navigate	25	90
antmaze-large-navigate	50	89
antmaze-large-navigate	100	58

Performance is non-monotone in k and the optimum scales with the underlying control horizon. On antmaze-large-navigate, success peaks at $k \in [25, 50]$ (around 90%) and degrades both for near-greedy ($k = 1$, 1%) and overly sparse subgoals ($k = 100$, 58%).

H Distillation and Q-Guidance can further enhance performance

We evaluate how Q-guidance improves goal-reaching performance and whether one-step distillation reduces inference cost while retaining the benefits of the guided teacher.

Performance improvements. Table 10 evaluates the joint effect of one-step distillation and Q-guidance on two stitching tasks, with results averaged over five random seeds. For D^{2}FM, average success increases from 71.5% to 93.2%, a gain of 21.7 percentage points. Specifically, success improves from 78.6% to 95.1% on pointmaze-medium-stitch and from 64.4% to 91.3% on antmaze-medium-stitch. HFIQL benefits less from the same stage: its success increases by 3.4 percentage points on PointMaze and 2.8 percentage points on AntMaze, yielding an average improvement of 3.1 percentage points. The larger gains for D^{2}FM suggest that distillation and Q-guidance are particularly effective when combined with dual occupancy weighting on these stitching tasks.

Figure 3 compares the base teacher (Teacher), the Q-guided teacher (Teacher+Q), and the Q-guided one-step student (Student+Q). Teacher+Q achieves the highest mean success, followed by Student+Q and Teacher. The improvement from Teacher to Teacher+Q supports the benefit of critic guidance. The distilled student retains most of the guided teacher's performance while outperforming the base teacher. Thus, one-step distillation retains part of the performance improvement provided by Q-guidance while enabling more efficient action generation.

Inference efficiency and training cost. Compared with Teacher+Q, Student+Q reduces end-to-end inference latency by 27.5–44.0%, corresponding to approximately 1.38–$1.78\times$ speedups. Most of the reduction comes from replacing iterative low-level flow integration with a single student forward pass. Training update time increases by approximately 9.3–9.9%. Together, these results demonstrate a performance–efficiency tradeoff: the guided teacher provides the highest success, while the distilled student offers faster inference and retains an advantage over the unguided teacher, at additional training cost.

I Runtime and memory usage

We compare the training cost of D^{2}FM against HIQL, HFIQL, HCFGRL, and FQL to contextualize the additional overhead introduced by hierarchical flow matching and dual-ratio correction. All numbers are measured on a single NVIDIA A800-SXM4-80GB (80 GB) with batch size 1024 on the antmaze-large-navigate task. Table 11 reports per-step training time and peak training-time GPU memory. D^{2}FM has the highest per-step training time in this comparison, reflecting its hierarchical flow actors together with additional ratio and value heads, but it stays close to its weighted-flow neighbour HFIQL (15.9 ms) and only about $1.3\times$ HIQL (13.4 ms). Peak GPU memory (1.51 GB) is also higher than the baselines (0.70–0.97 GB), but the absolute footprint remains

Table 10: Effect of one-step distillation and Q-guidance on policy performance. Success rates (%) are averaged over five random seeds.

Method	Configuration	PointMaze medium-stitch	AntMaze medium-stitch	Average
HFIQL	Without distill. & Q	64.6	54.0	59.3
	With distill. & Q	68.0	56.8	62.4
	Gain	+3.4	+2.8	+3.1
D^{2}FM	Without distill. & Q	78.6	64.4	71.5
	With distill. & Q	**95.1**	**91.3**	**93.2**
	Gain	+16.5	+26.9	+21.7

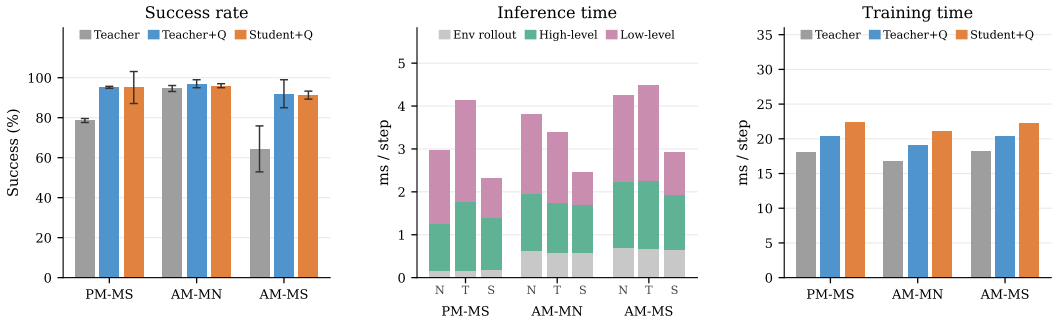


Figure 3: **Distillation: task performance, inference speedup, and training cost.** *Left:* success rate of the base teacher (Teacher), Q-guided teacher (Teacher+Q), and Q-guided one-step distilled student (Student+Q) on three tasks. *Middle:* normalized inference time (ms/step) decomposed into environment rollout, high-level subgoal selection, and low-level execution; N/T/S denote Teacher / Teacher+Q / Student+Q, respectively. *Right:* training update time (ms/step); PM/AM denotes pointmaze/antmaze; MN/MS denotes medium-navigate/medium-stitch.

negligible on an 80 GB GPU. Overall, the dual-ratio correction machinery incurs a moderate training overhead that is well within typical single-GPU research budgets.

Table 11: **Training cost.** Per-step training time and peak GPU memory on `antmaze-large-navigate` (single NVIDIA A800-SXM4-80GB (80 GB), batch 1024).

Method	Train time (ms/step)	Train mem. (GB)
HIQL	13.40	0.96
HFIQL	15.93	0.97
FQL	10.88	0.97
HCFGRL	4.96	0.70
D^{2}FM	17.53	1.51

J Hyperparameter sensitivity

The main hyperparameters in D^{2}FM are the high-level temperature β_{hi}, the low-level weighting strength α, and the Q-guidance coefficient λ_q (Eq. 33). Figure 4 sweeps $\beta_{hi} \in \{1, 3, 5, 10\}$ and $\alpha \in \{0.1, 0.2, 0.5, 1.0\}$ on `antmaze-medium-navigate`, and Table 13 sweeps $\lambda_q \in \{0.01, 0.1, 1.0, 10, 100\}$. The success-rate landscape is broad: across the β_{hi}–α grid, all 16 settings remain within 14 points of the best, and λ_q varies by less than 3 points across two orders of magnitude (0.01–1.0) before degrading at $\lambda_q = 100$. The default (β_{hi}=3, α=1.0, λ_q=0.1) sits well inside this stable region, indicating that D^{2}FM's gains are not driven by careful per-task tuning.

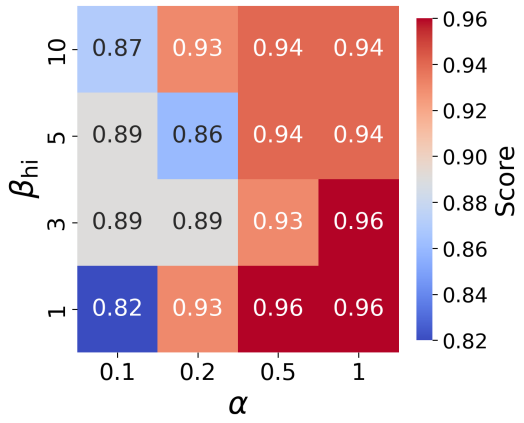


Figure 4: **Sensitivity to β_{hi} and α.** Average success rate on antmaze-medium-navigate across seeds.

Table 12: **Long-horizon OGBench maze tasks.** We report binary goal-reaching success (%) across seeds.

Environment	Dataset	Size	HIQL	OTA	D^{2}FM	D^{2}FM+OTA
HumanoidMaze	navigate	large	49 ± 4	83 ± 2	46 ± 2	83 ± 3
		giant	12 ± 4	92 ± 1	14 ± 3	99 ± 1
	stitch	large	28 ± 3	57 ± 3	30 ± 2	61 ± 2
		giant	3 ± 2	79 ± 3	4 ± 1	89 ± 3

Table 13: **Sensitivity to λ_q.** Average success rate (%) on antmaze-medium-navigate across seeds, with β_{hi} and α fixed at their defaults.

λ_q	0.01	0.1	1.0	10	100
Success (%)	95	96	94	88	46

K D^{2}FM ON LONG-HORIZON TASKS

OTA (Ahn et al., 2025) is a value-learning technique that uses option-aware temporal abstraction to improve hierarchical policy learning on long-horizon tasks by reducing the effective horizon of value backups. To explore the performance of D^{2}FM on such tasks, we integrate OTA-style value learning into our framework and denote the resulting variant as D^{2}FM+OTA. This combination retains our occupancy weighting for value learning and policy extraction while incorporating temporally abstracted value backups. OTA introduces an additional hyperparameter, the abstraction factor n, which controls the temporal span of these backups. We adopt the OTA hyperparameter settings reported by the original authors, including the abstraction factor, without additional tuning.

Table 12 reports results on large and giant HumanoidMaze navigation and stitching tasks. D^{2}FM+OTA improves over the base D^{2}FM on all four tasks. The largest gains occur in the giant layouts, where success rates increase from 14% to 99% on humanoidmaze-giant-navigate and from 4% to 89% on humanoidmaze-giant-stitch. Improvements also hold in the large layouts: navigation success rises from 46% to 83%, while stitching success increases from 30% to 61%. These results show that OTA-style value learning complements our occupancy-weighted framework and substantially improves performance on the evaluated long-horizon tasks.

L PROOFS

L.1 DISTRIBUTION-MATCHING AS OFFLINE GCRL

Proposition L.1 (Idealized distribution-matching view)**.** *For each goal g in the support of $p(g)$, define the reward-tilted state distribution:*

$$q_r(s;g) \coloneqq \frac{\exp(r(s;g)/\alpha)}{Z_r(g)}, \qquad Z_r(g) \coloneqq \int \exp(r(s;g)/\alpha)\, ds. \tag{34}$$

Then

$$-\alpha D_{\mathrm{KL}}(d_S^\pi(s;g)\|q_r(s;g)) + \alpha C_r = (1-\gamma)J(\pi) + \alpha \mathcal{H}(d_S^\pi(s;g)), \tag{35}$$

where $C_r \coloneqq \mathbb{E}_{g\sim p(g)}[\log Z_r(g)]$ is constant with respect to π.

Proof of Proposition L.1. We have that

$$\begin{aligned}
&\frac{(1-\gamma)}{\alpha} J(\pi) \\
=&\mathbb{E}_{g\sim p(g)}\mathbb{E}_{s\sim d_S^\pi(s;g)}\left[\log e^{r(s;g)/\alpha}\right] \\
=&\mathbb{E}_{g\sim p(g)}\mathbb{E}_{s\sim d_S^\pi(s;g)}\left[\log \frac{e^{r(s;g)/\alpha}}{Z_r(g)}\right] + \mathbb{E}_{g\sim p(g)}[\log Z_r(g)] \\
=&\mathbb{E}_{g\sim p(g)}\mathbb{E}_{s\sim d_S^\pi(s;g)}\left[\log \frac{q_r(s;g)}{d_S^\pi(s;g)}\right] + \mathbb{E}_{g\sim p(g)}\mathbb{E}_{d_S^\pi(s;g)}[\log d_S^\pi(s;g)] + C_r \\
=&\mathbb{E}_{g\sim p(g)}\left[-D_{\mathrm{KL}}(d_S^\pi(s;g)\|q_r(s;g)) - \mathcal{H}(d_S^\pi(s;g))\right] + C_r
\end{aligned} \tag{36}$$

Rearranging the equality gives the desired result. □

Remark L.2. Equation (35) gives an idealized bridge between control and distribution learning: maximizing an entropy-regularized goal-reaching objective is equivalent to matching the policy-induced state occupancy to a reward-tilted target distribution.

L.2 PROOF OF PROPOSITION 3.1

Proof. Recall that

$$p_\alpha(s;g) = \frac{d^O(s;g)\exp(r(s;g)/\alpha)}{Z_\alpha(g)}, \qquad Z_\alpha(g) = \int d^O(s;g)\exp(r(s;g)/\alpha)\, ds. \tag{37}$$

We have that

$$\begin{aligned}
&\frac{(1-\gamma)}{\alpha} J(\pi) \\
=&\mathbb{E}_{g\sim p(g)}\mathbb{E}_{s\sim d^\pi(s;g)}\left[\log e^{r(s;g)/\alpha}\right] \\
=&\mathbb{E}_{g\sim p(g)}\mathbb{E}_{s\sim d^\pi(s;g)}\left[\log \frac{d^O(s;g)\, e^{r(s;g)/\alpha}}{Z_\alpha(g)}\right] - \mathbb{E}_{g\sim p(g)}\mathbb{E}_{s\sim d^\pi(s;g)}\left[\log d^O(s;g)\right] + \mathbb{E}_{g\sim p(g)}[\log Z_\alpha(g)] \\
=&\mathbb{E}_{g\sim p(g)}\mathbb{E}_{s\sim d^\pi(s;g)}\left[\log \frac{p_\alpha(s;g)}{d^\pi(s;g)}\right] + \mathbb{E}_{g\sim p(g)}\mathbb{E}_{s\sim d^\pi(s;g)}\left[\log \frac{d^\pi(s;g)}{d^O(s;g)}\right] + \frac{C_\alpha}{\alpha} \\
=&\mathbb{E}_{g\sim p(g)}\left[-D_{\mathrm{KL}}\big(d^\pi(s;g)\|p_\alpha(s;g)\big) + D_{\mathrm{KL}}\big(d^\pi(s;g)\|d^O(s;g)\big)\right] + \frac{C_\alpha}{\alpha}.
\end{aligned} \tag{38}$$

Multiplying both sides by α and rearranging gives the per-goal identity

$$-\alpha\, \mathbb{E}_{g\sim p(g)}\big[D_{\mathrm{KL}}(d^\pi(s;g)\|p_\alpha(s;g))\big] + C_\alpha = (1-\gamma)J(\pi) - \alpha\, \mathbb{E}_{g\sim p(g)}\big[D_{\mathrm{KL}}(d^\pi(s;g)\|d^O(s;g))\big]. \tag{39}$$

By the data-processing inequality, the state-marginal KL is upper-bounded by the joint KL,

$$D_{\mathrm{KL}}\big(d^{\pi}(s;g)\|d^{O}(s;g)\big) \le D_{\mathrm{KL}}\big(d^{\pi}_{SA}(s,a;g)\|d^{O}_{SA}(s,a;g)\big), \tag{40}$$

and the joint KL is in turn upper-bounded by D_f under the assumption $D_f(P\|Q) \ge D_{\mathrm{KL}}(P\|Q)$, with equality when $f = \mathrm{KL}$. Substituting into (39) yields the desired inequality

$$\begin{aligned} &-\alpha\,\mathbb{E}_{g\sim p(g)}\big[D_{\mathrm{KL}}(d^{\pi}(s;g)\|p_{\alpha}(s;g))\big] + C_{\alpha} \\ &\quad \ge (1-\gamma)J(\pi) - \alpha\,\mathbb{E}_{g\sim p(g)}\big[D_f(d^{\pi}_{SA}(s,a;g)\|d^{O}_{SA}(s,a;g))\big]. \end{aligned} \tag{41}$$

□

L.3 Proof of Proposition 3.3

Proof. Introduce Lagrange multiplier $\nu(s;g)$ for each Bellman flow constraint in (6). The Lagrangian is

$$\begin{aligned} L(d_{SA},\nu) = \mathbb{E}_d[r(s;g)] - \alpha\, D_f(d_{SA}\|d^{O}_{SA}) \\ + \sum_{s,g}\nu(s;g)\Big[(1-\gamma)\mu_0(s) + \gamma\sum_{\tilde{s},\tilde{a}} T(s|\tilde{s},\tilde{a})d_{SA}(\tilde{s},\tilde{a};g) - \sum_{a} d_{SA}(s,a;g)\Big]. \end{aligned} \tag{42}$$

Rearranging the multiplier term produces a per-(s,a,g) inner expression $d(s,a;g)\,(\mathcal{T}\nu-\nu)(s,a;g) - \alpha\, d^{O}(s,a;g)\, f\big(d/d^{O}\big)$ together with the goal-averaged initial-state term $(1-\gamma)\mathbb{E}_{\mu_0,p(g)}[\nu]$. Maximizing over $d(s,a;g) \ge 0$ via the convex conjugate yields

$$\frac{d^*(s,a;g)}{d^{O}(s,a;g)} \;=\; \max\Big(0,\; (f')^{-1}\big((\mathcal{T}\nu^* - \nu^*)/\alpha\big)\Big), \tag{43}$$

and substituting back eliminates d and produces the unconstrained dual problem (7). □

L.4 Derivation of the state-ratio OPE dual (Eq. 10)

Proof. **Step 1: State-level Bellman flow.** The Bellman flow constraint in the primal (6) states that for any valid occupancy d^*:

$$\sum_{a} d^*_{SA}(s,a;g) = (1-\gamma)\mu_0(s) + \gamma\sum_{\bar{s},\bar{a}} T(s \mid \bar{s},\bar{a})\, d^*_{SA}(\bar{s},\bar{a};g), \quad \forall s,g. \tag{44}$$

The left-hand side is $d^*_S(s;g) = \sum_a d^*_{SA}(s,a;g)$. On the right-hand side, we substitute $d^*_{SA}(\bar{s},\bar{a};g) = d^{O}_{SA}(\bar{s},\bar{a};g)\, w^*_{SA}(\bar{s},\bar{a};g)$, where w^*_{SA} is the state-action occupancy ratio obtained from the dual (7). This gives:

$$d^*_S(s;g) = (1-\gamma)\mu_0(s) + \gamma\sum_{\bar{s},\bar{a}} d^{O}_{SA}(\bar{s},\bar{a};g)\, w^*_{SA}(\bar{s},\bar{a};g)\, T(s \mid \bar{s},\bar{a}) =: b(s;g). \tag{45}$$

Note that d^*_S appears only on the left; $b(s;g)$ is fully determined by the given w^*_{SA} and the dataset.

Step 2: Lagrangian formulation. Step 1 already pins $d^*_S(s;g) = b(s;g)$ pointwise, so the recovery $w_S(s;g) \coloneqq d^*_S(s;g)/d^{O}_S(s;g) = b(s;g)/d^{O}_S(s;g)$ is unique. The reason we go through a Lagrangian is computational rather than statistical: $b(s;g)$ involves an expectation over the transition kernel and over w^*_{SA} that we cannot evaluate in closed form, so we expose it through a dual variable that admits a sample-based estimator. Concretely, we introduce the auxiliary program

$$\min_{w_S\ge 0} \sum_{s} d^{O}_S(s;g)\, f(w_S(s;g)) \quad \text{s.t.} \quad w_S(s;g)\, d^{O}_S(s;g) = b(s;g), \;\; \forall s,g, \tag{46}$$

whose unique feasible point is $w_S = b/d^{O}_S$ regardless of the choice of f; the convex generator f is what enables the dual derivation that follows.

Step 3: Lagrangian. Introduce a Lagrange multiplier $U(s;g)$ for each constraint $b(s;g) - w_S(s;g)\, d_S^O(s;g) = 0$:

$$\begin{aligned}\mathcal{L}(w_S, U) &= \sum_s d_S^O(s;g)\, f(w_S(s;g)) + \sum_s U(s;g)\Big[b(s;g) - w_S(s;g)\, d_S^O(s;g)\Big] \\ &= \sum_s d_S^O(s;g)\Big[f(w_S(s;g)) - U(s;g)\, w_S(s;g)\Big] + \sum_s U(s;g)\, b(s;g). \end{aligned} \tag{47}$$

Expanding $b(s;g)$:

$$\begin{aligned}\sum_s U(s;g)\, b(s;g) &= (1-\gamma)\sum_s U(s;g)\,\mu_0(s) \\ &\quad + \gamma \sum_s U(s;g) \sum_{\bar{s},\bar{a}} d^O(\bar{s},\bar{a};g)\, w_{SA}(\bar{s},\bar{a};g)\, T(s \mid \bar{s},\bar{a}) \\ &= (1-\gamma)\,\mathbb{E}_{s_0\sim\mu_0}[U(s_0;g)] \\ &\quad + \gamma\, \mathbb{E}_{(\bar{s},\bar{a})\sim d^O}\Big[w_{SA}(\bar{s},\bar{a};g)\; \mathbb{E}_{s'\sim T(\cdot\mid\bar{s},\bar{a})}[U(s';g)]\Big] \\ &= (1-\gamma)\,\mathbb{E}_{s_0\sim\mu_0}[U(s_0;g)] + \gamma\, \mathbb{E}_{(s,a,s')\sim d^O}[w_{SA}(s,a;g)\cdot U(s';g)]. \end{aligned} \tag{48}$$

The second equality uses the empirical $(s,a,s') \sim d^O$ tuples as an unbiased estimator of $\mathbb{E}_{(s,a)\sim d^O}\, \mathbb{E}_{s'\sim T}[U(s';g)]$. So the full Lagrangian is:

$$\mathcal{L}(w_S, U) = \mathbb{E}_{s\sim d_S^O}\big[f(w_S) - U\, w_S\big] + (1-\gamma)\,\mathbb{E}_{\mu_0}[U] + \gamma\,\mathbb{E}_{d^O}[w_{SA}\cdot U(s')]. \tag{49}$$

Step 4: Inner minimization. The dual is $\max_U \min_{w_S\ge 0} \mathcal{L}$. For each state s, the inner problem is:

$$\min_{w\ge 0}\; f(w) - U\, w. \tag{50}$$

Taking the derivative and setting to zero: $f'(w) = U$, so $w^* = (f')^{-1}(U)$. With the constraint $w \ge 0$: $w^* = \max(0, (f')^{-1}(U))$. The minimum value is $f(w^*) - Uw^* = -f_p^*(U)$, where $f_p^*(u) \coloneqq \sup_{x\ge 0}[ux - f(x)]$ is the positive conjugate.

Step 5: Dual objective. Substituting back:

$$\max_U\; -\mathbb{E}_{s\sim d_S^O}[f_p^*(U)] + (1-\gamma)\,\mathbb{E}_{\mu_0}[U] + \gamma\,\mathbb{E}_{d^O}[w_{SA}\cdot U(s')]. \tag{51}$$

Flipping the sign to a minimization:

$$\min_U\; \mathbb{E}_{s\sim d_S^O}[f_p^*(U(s;g))] - (1-\gamma)\,\mathbb{E}_{\mu_0}[U(s_0;g)] - \gamma\,\mathbb{E}_{d^O}[w_{SA}(s,a;g)\cdot U(s';g)], \tag{52}$$

which is (10). The recovered state ratio is $w_S^*(s;g) = \max(0, (f')^{-1}(U^*(s;g)))$ from Step 4. □

L.5 Proof of Proposition L.3

Proposition L.3 (Conditional consistency of weighted flow matching)**.** *Let g denote the conditioning subgoal in equation 15, and let $q_{\mathrm{lo}}(s,a,g)$ denote the population distribution of its training triples. For a fixed nonnegative weight w_{SA}, define*

$$Z_w(s,g) = \mathbb{E}_{a\sim q_{\mathrm{lo}}(\cdot\mid s,g)}[w_{SA}(s,a;g)], \tag{53}$$

$$\pi_w(a \mid s,g) = \frac{q_{\mathrm{lo}}(a \mid s,g)\, w_{SA}(s,a;g)}{Z_w(s,g)}, \tag{54}$$

where π_w is defined whenever $Z_w(s,g) > 0$.

Assume that

$$0 < \mathbb{E}_{q_{\mathrm{lo}}}\big[w_{SA}(s,a;g)(1+\|a\|^2)\big] < \infty,$$

the velocity model can represent the population CFM vector field, and the associated ODE and continuity equation satisfy the existence and uniqueness conditions for flow transport. Then, at a global minimum of the population objective equation 15, exact ODE integration from Gaussian noise generates $\pi_w(\cdot \mid s,g)$ for $q_{\mathrm{lo}}(s,g)$-almost every conditioning pair with $Z_w(s,g) > 0$.

In particular, let d^*_{SA} be an optimizer of equation 6, and assume that $d^O_{SA}(s,a;g) > 0$ wherever $d^*_{SA}(s,a;g) > 0$. If

$$w_{SA}(s,a;g) = \frac{d^*_{SA}(s,a;g)}{d^O_{SA}(s,a;g)} \quad \textit{and} \quad q_{\mathrm{lo}}(a \mid s,g) = \frac{d^O_{SA}(s,a;g)}{d^O_S(s;g)},$$

then, on the above conditioning pairs with $d^*_S(s;g) > 0$,

$$\pi_w(a \mid s,g) = \pi^{\mathrm{lo}*}(a \mid s,g) \coloneqq \frac{d^*_{SA}(s,a;g)}{d^*_S(s;g)}.$$

Remark L.4. The proposition characterizes the conditional distribution learned by the occupancy-weighted low-level teacher. Identifying this distribution with $\pi^{\mathrm{lo}*}$ requires consistency between the training action conditional and the offline conditional used to define the occupancy ratio. Including the same subgoals in the relabeling support alone does not establish this condition. Proposition L.5 provides a complementary approximation-error bound for state-weighted value regression with a fixed target. These results characterize the two weighted learning objectives, without establishing a value fixed point or optimality of the composed hierarchical policy.

Proof. Let

$$\ell_\theta(s,a,g) = \mathbb{E}_{\sigma,x^0}\left[\left\| v_{\theta_{\mathrm{lo}}}(\sigma,s,x^\sigma;g) - (a - x^0)\right\|^2\right],$$

where $x^0 \sim \mathcal{N}(0,I)$, $\sigma \sim \mathrm{Unif}([0,1])$, and $x^\sigma = (1-\sigma)x^0 + \sigma a$. By the definitions of Z_w and π_w,

$$\mathcal{L}_{\mathrm{lo}}(\theta) = \mathbb{E}_{(s,a,g)\sim q_{\mathrm{lo}}}\left[w_{SA}(s,a;g)\,\ell_\theta(s,a,g)\right] \tag{55}$$

$$= \mathbb{E}_{(s,g)\sim q_{\mathrm{lo}}}\left[Z_w(s,g)\,\mathbb{E}_{a\sim\pi_w(\cdot|s,g)}[\ell_\theta(s,a,g)]\right]. \tag{56}$$

Conditioning pairs with $Z_w(s,g) = 0$ contribute nothing to the objective. For every remaining pair, $Z_w(s,g)$ is a positive factor independent of the velocity model. Under the realizability assumption, the population minimizer therefore agrees almost everywhere with

$$v^\star_w(\sigma,s,x;g) = \mathbb{E}\left[a - x^0 \,\middle|\, (1-\sigma)x^0 + \sigma a = x,\ s,g\right],$$

where $a \sim \pi_w(\cdot \mid s,g)$ and $x^0 \sim \mathcal{N}(0,I)$ are independent. This is the marginal CFM vector field associated with the linear interpolation between Gaussian noise and $\pi_w(\cdot \mid s,g)$. By the flow-matching transport result (Lipman et al., 2023) and the stated regularity assumptions, exact integration of this field generates $\pi_w(\cdot \mid s,g)$.

Under the additional occupancy-ratio and conditional-distribution assumptions,

$$q_{\mathrm{lo}}(a \mid s,g)\, w_{SA}(s,a;g) = \frac{d^*_{SA}(s,a;g)}{d^O_S(s;g)}.$$

Integrating over actions gives $Z_w(s,g) = d^*_S(s;g)/d^O_S(s;g)$. Normalizing consequently yields

$$\pi_w(a \mid s,g) = \frac{d^*_{SA}(s,a;g)}{d^*_S(s;g)},$$

which proves the final claim. □

L.6 Proof of Proposition L.5

Proposition L.5 (Approximation error of state-weighted expectile regression)**.** *Fix a target value function $\bar{V}$ and $\tau \in [1/2,1)$. Suppose the population distribution underlying (12) factorizes as*

$$d^O(s,s',g) = p(g)\, d^O_S(s;g)\, p^O(s' \mid s,g).$$

For $p(g)$-almost every goal g, assume that $d^O_S(s;g) > 0$ wherever $d^*_S(s;g) > 0$. Define the fixed backup target

$$Y_{\bar{V}}(s,s',g) = r(s;g) + \gamma \bar{V}(s';g),$$

and assume that it is square-integrable under $p(g)d^*_S(s;g)p^O(s' \mid s,g)$.

Let the population weighted regression loss be

$$\mathcal{L}_w(V;\bar{V}) = \mathbb{E}_{(s,s',g)\sim d^O}\left[w_S(s;g)L_2^\tau(Y_{\bar{V}}(s,s',g) - V(s;g))\right],$$

and define the conditional expectile target

$$V^\dagger(s;g) = \arg\min_{v\in\mathbb{R}} \mathbb{E}_{s'\sim p^O(\cdot|s,g)}\left[L_2^\tau(Y_{\bar{V}}(s,s',g) - v)\right].$$

Write

$$\|V\|_{d_S^*}^2 = \mathbb{E}_{g\sim p(g),\, s\sim d_S^*(\cdot;g)}\left[V(s;g)^2\right].$$

Then, for every V with $\|V\|_{d_S^*} < \infty$,

$$\begin{aligned}(1-\tau)\|V - V^\dagger\|_{d_S^*}^2 &\le \mathcal{L}_w(V;\bar{V}) - \mathcal{L}_w(V^\dagger;\bar{V}) \\ &\le \tau\|V - V^\dagger\|_{d_S^*}^2. \end{aligned} \tag{57}$$

Furthermore, let $\mathcal{F}$ be a nonempty class of value functions with finite $\|\cdot\|_{d_S^*}$ norm. If $\widehat{V}\in\mathcal{F}$ satisfies

$$\mathcal{L}_w(\widehat{V};\bar{V}) \le \inf_{V\in\mathcal{F}} \mathcal{L}_w(V;\bar{V}) + \epsilon, \qquad \epsilon \ge 0,$$

then

$$\|\widehat{V} - V^\dagger\|_{d_S^*}^2 \le \frac{\tau}{1-\tau}\inf_{V\in\mathcal{F}}\|V - V^\dagger\|_{d_S^*}^2 + \frac{\epsilon}{1-\tau}. \tag{58}$$

Proof. Exact state weights give the change-of-measure identity

$$\mathcal{L}_w(V;\bar{V}) = \mathbb{E}_{\substack{g\sim p(g),\, s\sim d_S^*(\cdot;g)\\ s'\sim p^O(\cdot|s,g)}}\left[L_2^\tau(Y_{\bar{V}}(s,s',g) - V(s;g))\right].$$

For fixed (s,g), define the conditional risk

$$R_{s,g}(v) = \mathbb{E}_{s'\sim p^O(\cdot|s,g)}\left[L_2^\tau(Y_{\bar{V}}(s,s',g) - v)\right].$$

As a function of v, the expectile loss $L_2^\tau(y-v)$ is $2(1-\tau)$-strongly convex and has a 2τ-Lipschitz derivative. These properties are preserved by conditional expectation. Since $V^\dagger(s;g)$ minimizes $R_{s,g}$, its first derivative vanishes there. Consequently,

$$\begin{aligned}(1-\tau)|v - V^\dagger(s;g)|^2 &\le R_{s,g}(v) - R_{s,g}(V^\dagger(s;g)) \\ &\le \tau|v - V^\dagger(s;g)|^2. \end{aligned}$$

Integrating over $p(g)d_S^*(s;g)$ proves equation 57.

Applying its lower bound to $\widehat{V}$, followed by the assumed optimization guarantee and its upper bound, yields

$$\begin{aligned}(1-\tau)\|\widehat{V} - V^\dagger\|_{d_S^*}^2 &\le \mathcal{L}_w(\widehat{V};\bar{V}) - \mathcal{L}_w(V^\dagger;\bar{V}) \\ &\le \inf_{V\in\mathcal{F}}\left[\mathcal{L}_w(V;\bar{V}) - \mathcal{L}_w(V^\dagger;\bar{V})\right] + \epsilon \\ &\le \tau\inf_{V\in\mathcal{F}}\|V - V^\dagger\|_{d_S^*}^2 + \epsilon. \end{aligned}$$

Dividing by $1-\tau$ completes the proof. □