BEYOND SAFE DEMONSTRATIONS: FAILURE-INFORMED FLOW POLICIES FOR OFFLINE SAFE REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Can an offline learner satisfy a deployment cost budget without complete trajectories whose cumulative cost is within that budget? Bootstrapped cost estimates may fail to capture delayed costs that only become visible in recorded subtrajectories. We introduce FIBA-Flow, a failure-informed, budget-adaptive approach to offline safe reinforcement learning. For each candidate action, we combine a cost-to-go estimate with a short-horizon cumulative-cost prediction to obtain a cost score. We optimize target action probabilities to maximize entropy-regularized predicted return under a remaining-budget constraint on the mean cost score. A budget-conditioned flow is trained to match these target distributions for one-step action generation. A guide learns to adjust the flow's internal budget input through action-distribution matching. Its targets are constructed across budget conditions under a constraint on the predicted cost set by the actual remaining budget, using critics that account for subsequent guide decisions. Evaluations on 38 DSRL tasks show that FIBA-Flow satisfies the cost constraint on every task and achieves the highest normalized return on 25 tasks among budget-compliant methods. Our anonymous link is: https://anonymous.4open.science/r/paper-6d0be256/

1 INTRODUCTION

Offline reinforcement learning (RL) learns policies from a fixed dataset without additional environment interaction during training (Park et al., 2025; Nguimatsia Tiofack et al., 2026; Chemingui et al., 2026; Levine et al., 2020). In safety-critical control, the objective is to maximize expected return subject to a bound on expected cumulative cost (Xue et al., 2026; Wachi et al., 2026; Cao et al., 2025; Gong et al., 2025; Liu et al., 2025; Koirala et al., 2025). Yet complete trajectories satisfying the deployment budget may be scarce or absent in the available data (Yao et al., 2024; Burnwal et al., 2026). This raises a practical question: *how can an offline learner use such data to learn a high-return policy whose expected cumulative cost satisfies the deployment budget?*

Trajectories that violate the deployment budget can still provide useful supervision for cost prediction. Cost critics estimate the expected cost-to-go under the policy being learned, but during temporal-difference (TD) learning, estimates at earlier decision points may not yet reflect costs occurring several steps later. Recorded short subtrajectories provide cumulative-cost targets without bootstrapping, but these targets reflect subsequent actions recorded in the dataset rather than actions generated by the learned policy. The challenge is to combine these two sources of cost information with return estimates to guide action selection under a remaining-budget constraint.

We introduce FIBA-Flow, a failure-informed, budget-adaptive approach that trains a flow policy to match target action distributions constructed under a remaining-budget constraint on the mean cost score. Candidate cost scores combine cost-to-go estimates with learned short-horizon cumulative-cost predictions. We determine target action probabilities by maximizing entropy-regularized predicted return subject to a remaining-budget constraint on the probability-weighted mean cost score. As illustrated in Figure 1(b,c), incorporating short-horizon cost predictions shifts the target action probabilities toward the candidate that avoids the nearby hazard. The budget-conditioned flow is trained to approximate these target distributions for one-step action generation (Wang et al., 2026;

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.

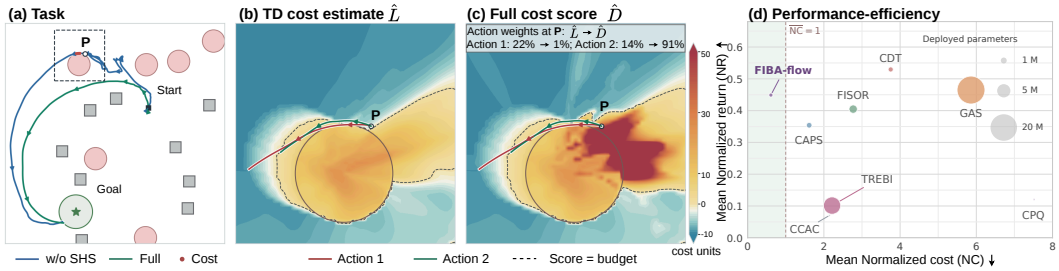

Figure 1: (a) In PointGoal2, the policy without short-horizon scoring (w/o SHS) incurs cost near a hazard, whereas the full policy reaches the goal without cost. (b,c) At state P, the TD cost estimate for *Action 1* lies below the budget. Incorporating short-horizon cost predictions raises its score above the budget and shifts the flow-training target toward *Action 2*. (d) Mean normalized return and normalized cost over the 38 tasks in Table 1. Bubble area indicates the number of parameters.

Zhan et al., 2026; Espinosa-Dice et al., 2026), without candidate scoring or probability optimization at deployment (Zhang et al., 2025; Brahmanage & Kumar, 2026).

Even after training on budget-constrained targets, a smaller internal budget scale need not produce actions with lower predicted cost. The actual remaining budget specifies the cost constraint, while each budget scale is evaluated using actions generated by the base policy at that scale. Motivated by learned steering of generative policies (Ki et al., 2026; Zhang et al., 2026), we introduce a guide that learns to select this scale while keeping the actual budget constraint unchanged. We train the guide to bring the action distribution at its selected scale closer to targets that favor high predicted return under that constraint. Since future actions also depend on the guide, we construct its training targets using critics that estimate return and cost under the guided policy.

We evaluate FIBA-Flow on 38 DSRL tasks (Liu et al., 2023a) at cost budget $B = 10$, after removing all complete training trajectories satisfying that budget. FIBA-Flow satisfies the cost constraint on all 38 tasks and achieves the highest normalized return on 25 tasks among the evaluated budget-compliant methods (Table 1). Ablations further show that short-horizon cost scoring and adaptive budget conditioning reduce cost on most tasks, supporting their contributions to cost control.

2 Related Work

Offline safe reinforcement learning. Early work addressed distribution shift and cost constraints through cost penalties for out-of-distribution actions (Xu et al., 2022) and stationary-distribution correction with conservative cost estimates (Lee et al., 2022). A reachability-based formulation separates reward maximization in feasible regions from risk reduction elsewhere (Zheng et al., 2024). DRCORL (Guo et al., 2025a) regularizes a Gaussian actor with a diffusion behavior model and handles conflicting reward and cost gradients. Budget-conditioned policies span sequence modeling (Liu et al., 2023b) and actor–critic learning (Guo et al., 2025b), while TREBI (Lin et al., 2023) formulates budget-constrained control over trajectory distributions and implements it through guided diffusion planning. SDGD (Chen et al., 2026) combines budget conditioned classifier-free guidance with reward-gradient refinement, using feasible trajectory relabeling to discourage reward-induced cost increases. SafeFQL couples reachability-based safety evaluation with a distilled one-step actor (Tayal et al., 2026). B2R realigns cost-to-go conditioning to represent a safety boundary (Su et al., 2026). FIBA-Flow trains short-horizon cost predictors on cumulative costs from recorded subtrajectories without bootstrapping. Their predictions are combined with cost-to-go estimates to construct target action distributions under a remaining-budget constraint on the mean cost score.

Generative policies. Generative policies combine expressive action distributions with value-based improvement through diffusion (Wang et al., 2023) and flow-based behavior regularization (Park et al., 2025; Nguimatsia Tiofack et al., 2026). Guidance weights can be learned through distribution matching (Galashov et al., 2026) or optimized through a stochastic optimal control formulation of diffusion guidance (Azangulov et al., 2025). Negative guidance provides another mechanism for shaping generation (Kim et al., 2026). FIBA-Flow uses MeanFlow (Geng et al., 2026) for one-step action generation and learns a separate guide to select its internal budget condition at each environment step. The guide is trained to match action mixtures constructed under a mean predicted-

cost constraint set by the actual remaining budget. To construct these mixtures, we use return and cost estimates that account for the guide's choice of budget scale at subsequent steps.

3 PRELIMINARIES

3.1 OFFLINE SAFE REINFORCEMENT LEARNING

We consider a finite-horizon constrained Markov decision process with reward r, nonnegative cost c, and time limit T (Xu et al., 2022). For a policy π, define the discounted return $J_R(\pi) = \mathbb{E}_{\tau\sim\pi}[\sum_{t=0}^{T-1}\gamma^t r_t]$ and undiscounted cost $J_C(\pi) = \mathbb{E}_{\tau\sim\pi}[\sum_{t=0}^{T-1} c_t]$, where γ is the reward discount factor. Using only a fixed offline dataset $\mathcal{D}$, we seek a policy under cost budget B:

$$\max_{\pi} J_R(\pi) \quad \text{s.t.} \quad J_C(\pi) \le B. \tag{1}$$

At environment step $t \in \{0, \dots, T-1\}$, the context is $z = (h, T-t, \beta, B)$, where h is the causal history, and β the signed remaining budget: $\beta = B - \sum_{k=0}^{t-1} c_k$. The action values $Q_R^\pi(z,a)$ and $Q_C^\pi(z,a)$ denote the expected discounted return and undiscounted cumulative cost after taking a and subsequently following π, respectively. Hats denote learned estimates.

3.2 FLOW MATCHING

Conditioned on z, flow matching learns a velocity field that transports Gaussian noise to a target action distribution (Lipman et al., 2022). Draw an action a from this target and independent noise $\epsilon \sim \mathcal{N}(0, I)$. For generation time $s \in (0,1)$, define the linear interpolation and marginal velocity:

$$X_s = (1-s)a + s\epsilon, \qquad v(z,x,s) = \mathbb{E}[\epsilon - a \mid X_s = x, z]. \tag{2}$$

Conditional flow matching estimates this field by regressing a velocity predictor toward $\epsilon - a$. Sampling integrates the learned field from $s = 1$ (noise) to $s = 0$ (target).

MeanFlow (Geng et al., 2026) models the average velocity along a trajectory of v over a generation-time interval $0 \le r < s \le 1$. Its velocity network $u_\theta(z, x, r, s)$ predicts the displacement over $[r, s]$, divided by $s - r$, along a trajectory passing through x at time s. Training uses an identity between average and instantaneous velocities. Setting $(r, s) = (0, 1)$ defines the base policy π_θ:

$$a = \epsilon - u_\theta(z, \epsilon, 0, 1) \sim \pi_\theta(\cdot \mid z), \qquad \epsilon \sim \mathcal{N}(0, I). \tag{3}$$

Generation times r, s are distinct from environment time t. Section 4.2 specifies the weighted training objective and MeanFlow regression target.

4 METHOD

Although the data contain no complete budget-compliant trajectories, individual transitions and short subtrajectories may still provide useful cost information. We combine cost-to-go estimates with short-horizon cost predictions to score candidate actions, then optimize their probabilities for entropy-regularized predicted return under a constraint on the mean cost score (Section 4.1). A budget-conditioned MeanFlow policy learns the resulting target action distributions for one-step action generation (Section 4.2). A guide selects the internal budget scale and is trained to make the generated action distribution approximate targets constrained by the actual remaining budget (Section 4.3). Figure 2 summarizes target construction, flow learning, and adaptive budget guidance.

4.1 FAILURE-INFORMED BUDGET-CONSTRAINED TARGETS

At state P in Figure 1(b,c), the TD cost estimate for *Action 1* lies below the budget. Incorporating short-horizon cost predictions raises its score above the budget and shifts the target action probabilities toward *Action 2*. Budget-violating trajectories can reveal costly behavior that TD estimates understate. We incorporate this short-horizon evidence into the budget-constrained target distribution, allowing failures to inform which actions the policy learns to favor.

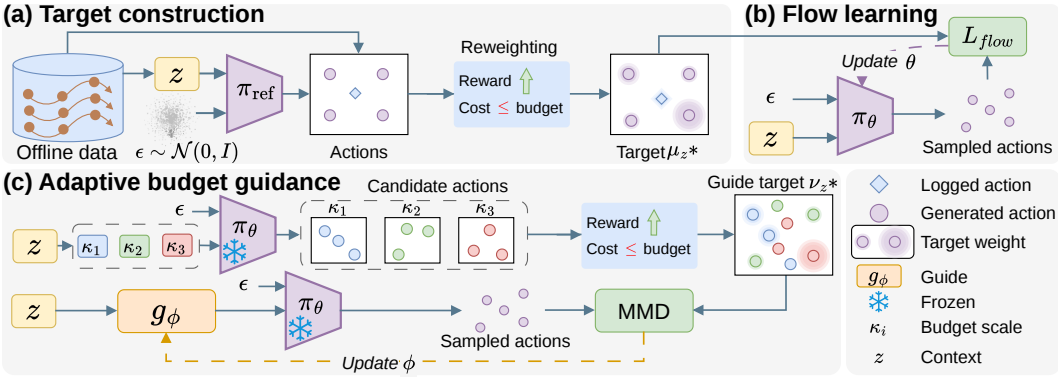

Figure 2: Overview of FIBA-Flow. (a) Actions generated by a reference policy π_{ref} and the logged action are reweighted according to reward and the cost budget to form μ_z^*. (b) The flow loss L_{flow} trains a policy π_θ to match μ_z^*. (c) With π_θ frozen, candidate actions generated at different budget scales κ_i are reweighted to form the guide target ν_z^*. The guide g_ϕ is trained to adjust the budget scale by minimizing a MMD loss computed from the sampled actions and ν_z^*.

For a transition sampled from $\mathcal{D}$ with context z, we combine its logged action with $N-1$ candidate actions from the reference policy π_{ref} fitted to $\mathcal{D}$. For candidate a_i, let $\widehat{R}_i$ and $\widehat{L}_i$ estimate return-to-go and cost-to-go under the base policy π_θ. Let $\widehat{M}_i$ predict the cost over a fixed window of H recorded steps, truncated at episode termination. We define the cost score:

$$\widehat{D}_i = \max\left\{\widehat{L}_i,\ \min_j \widehat{M}_j + \max\left(1, \frac{T-t}{H}\right)\left(\widehat{M}_i - \min_j \widehat{M}_j\right)\right\}. \tag{4}$$

When $T-t \le H$, the short-horizon term is $\widehat{M}_i$. For $T-t > H$, this term keeps $\min_j \widehat{M}_j$ unchanged and multiplies each candidate's difference $\widehat{M}_i - \min_j \widehat{M}_j$ by $(T-t)/H$. The maximum in Eq. 4 ensures that $\widehat{D}_i \ge \widehat{L}_i$. This score need not upper-bound the true remaining cost.

We choose action probabilities by maximizing regularized predicted return:

$$w^* = \arg\max_{w \in \Delta_N} \sum_{i=1}^N w_i \widehat{R}_i - \eta\, \mathrm{KL}\big(w \,\|\, \mathrm{Unif}(N)\big) \quad \text{s.t.} \quad \sum_{i=1}^N w_i \widehat{D}_i \le [\beta]_+. \tag{5}$$

Here Δ_N is the probability simplex, $\mathrm{Unif}(N)$ is uniform over the candidate actions, $\eta > 0$, and $[\beta]_+ = \max(\beta, 0)$. The constraint bounds the mean score, so actions with $\widehat{D}_i > [\beta]_+$ may retain positive probability. If $\min_i \widehat{D}_i > [\beta]_+$, we maximize the same objective over minimum-score candidates; this fallback remains infeasible for Eq. 5.

With candidates and scores fixed, the feasible problem has a unique optimizer. For $[\beta]_+ > \min_i \widehat{D}_i$, the optimizer is $w^* = w(\lambda^*)$, where the weights and the derivative of the mean cost score are:

$$w_i(\lambda) = \frac{\exp[(\widehat{R}_i - \lambda \widehat{D}_i)/\eta]}{\sum_k \exp[(\widehat{R}_k - \lambda \widehat{D}_k)/\eta]}, \qquad \frac{\mathrm{d}}{\mathrm{d}\lambda} \sum_i w_i(\lambda) \widehat{D}_i = -\frac{1}{\eta} \mathrm{Var}_{i \sim w(\lambda)}[\widehat{D}_i] \le 0. \tag{6}$$

If $w(0)$ meets the budget, $\lambda^* = 0$. Otherwise, bisection finds the unique crossing of $[\beta]_+$. The remaining budget thus sets the cost penalty for each candidate set (Appendix A.1).

The resulting weights, including the fallback weights, define the target action distribution

$$\mu_z^* = \sum_{i=1}^N w_i^* \delta_{a_i}, \tag{7}$$

where δ_a is a point mass at action a. This target varies with the sampled candidate pool; we suppress that randomness in its notation. Section 4.2 uses μ_z^* to supervise the budget-conditioned policy.

4.2 Constrained-Target Flow Learning

We train π_θ by applying MeanFlow to the target action distribution μ_z^*, using the interpolation and marginal velocity defined in Section 3.2. We draw Gaussian noise ϵ independently of the candidate

actions and set $x_{i,s} = (1-s)a_i + s\epsilon$. For candidate a_i, the regression target for u_θ is:

$$u_i^{\mathrm{tgt}} = v - (s-r)\big(\partial_s u_\theta + (\nabla_x u_\theta)v\big), \tag{8}$$

where v is evaluated at $(z, x_{i,s}, s)$, and u_θ and its derivatives at $(z, x_{i,s}, r, s)$. Generation times $0 < r < s < 1$ are sampled independently of the candidates and noise, conditional on z. The MeanFlow loss weights each squared regression error by w_i^*:

$$\mathcal{L}_{\mathrm{flow}}(\theta) = \mathbb{E}\left[\sum_{i=1}^{N} w_i^* \left\| u_\theta(z, x_{i,s}, r, s) - \mathrm{sg}(u_i^{\mathrm{tgt}}) \right\|_2^2\right]. \tag{9}$$

Here sg stops gradients. The expectation includes contexts, candidate sampling, noise, and generation times; candidate actions and weights are held fixed during each update. The weights therefore shape both the velocity field in the regression target and each action's contribution to the loss.

Proposition 1 connects Eq. 9 to the target action distributions in Section 4.1.

Proposition 1 (Consistency with averaged action targets) *Fix z and all target-generating components. Under the sampling scheme above and the regularity and endpoint conditions in Appendix A.1, any function-space stationary point of the stop-gradient MeanFlow update satisfies:*

$$\pi_\theta(\cdot \mid z) = \mathbb{E}[\mu_z^* \mid z], \tag{10}$$

where the expectation is over candidate construction.

For each candidate action set, the target's mean score is at most $[\beta]_+$ when feasible and equals the minimum candidate score under fallback. The base policy's mean predicted cost is therefore bounded by $[\beta]_+$ plus the expected positive excess of the minimum candidate score above this budget. At inference, the base policy generates an action with a single flow-network evaluation (Eq. 3); candidate construction, critic evaluation, and reweighting are confined to training.

4.3 Policy-Consistent Budget Adaptation

At each environment step, a guide g_ϕ selects an internal budget scale $\kappa = g_\phi(z) > 0$ for the base policy π_θ. We train the guide through action distribution matching to targets that favor predicted return under a mean predicted-cost constraint set by the actual remaining budget. We define the adjusted context z_κ and the guided policy π as

$$z_\kappa = \big(h, T-t, \beta + (\kappa-1)B, \kappa B\big), \qquad \pi(\cdot \mid z) = \pi_\theta(\cdot \mid z_\kappa). \tag{11}$$

The range of κ is specified in Table 3 in the Appendix. This transformation preserves the incurred cost $B - \beta$, while the actual budget B and remaining budget β remain unchanged. During target construction, z_κ controls action generation, but actions are evaluated at the actual context z, with the mean predicted-cost constraint set by $[\beta]_+$. Setting $\kappa = 1$ recovers the base policy.

Since future actions also use the guide, we evaluate candidate actions assuming that subsequent decisions follow π. We therefore learn separate reward and cost critics $\widehat{Q}_R^\pi$ and $\widehat{Q}_C^\pi$, distinct from those for the base policy in Section 4.1. For a transition with cost c and next history h', the next actual context is $z' = (h', T-t-1, \beta - c, B)$. The cost-to-go satisfies:

$$Q_C^\pi(z,a) = \mathbb{E}[c + Q_C^\pi(z', a') \mid z, a], \qquad a' \sim \pi(\cdot \mid z'). \tag{12}$$

The continuation value is zero at termination. The guide is evaluated again at z' to generate a'. The reward relation uses the same next action with discount γ.

Larger budget scales need not yield higher predicted costs, so we compare conditions through the actions they generate. For J candidate scales κ_j, including 1, we generate M_G actions a_{jm} per condition using Eq. 3 with the target copy $\pi_{\bar\theta}$ of π_θ at z_{κ_j}. We reuse the same M_G independent Gaussian draws across conditions and evaluate the resulting actions at the actual context z, giving the mean predicted return $\bar R_j$ and mean predicted cost $\bar C_j$ for each candidate scale:

$$\bar R_j = \frac{1}{M_G}\sum_{m=1}^{M_G} \widehat{Q}_R^\pi(z, a_{jm}), \qquad \bar C_j = \frac{1}{M_G}\sum_{m=1}^{M_G} \widehat{Q}_C^\pi(z, a_{jm}). \tag{13}$$

We construct the guide's training target by assigning probabilities $p = (p_1, \ldots, p_J)$ to the candidate budget conditions, where p_j is the weight of condition κ_j. Let $\widetilde{R}_j$ denote the normalized return scores obtained from $\bar{R}_j$ using Eq. 26. As in Eq. 5, we solve for the optimal probabilities p^*:

$$p^* = \arg\max_{p \in \Delta_J} \sum_j p_j \widetilde{R}_j - \eta_G \, \mathrm{KL}\big(p \| \, \mathrm{Unif}(J)\big) \quad \text{s.t.} \quad \sum_j p_j \bar{C}_j \le [\beta]_+. \tag{14}$$

Here Δ_J is the probability simplex, $\mathrm{Unif}(J)$ is uniform over the candidate conditions, and $\eta_G > 0$ controls regularization. The constraint uses mean predicted costs under the guided policy directly, rather than the cost scores in Eq. 4. If $\min_j \bar{C}_j > [\beta]_+$, we maximize the same objective over distributions supported on conditions with the minimum mean predicted cost; this fallback remains infeasible for Eq. 14. We use p^* for both cases and define the guide target distribution as:

$$\nu_z^* = \sum_{j=1}^{J} \sum_{m=1}^{M_G} \frac{p_j^*}{M_G} \, \delta_{a_{jm}}. \tag{15}$$

Each condition contributes total probability mass p_j^*, shared equally among its M_G sampled actions.

Averaging the candidate scales as $\sum_j p_j^* \kappa_j$ does not generally produce the corresponding mixture of action distributions. We therefore train g_ϕ to approximate ν_z^* through the action distribution induced by its selected condition, using squared maximum mean discrepancy (MMD):

$$\mathcal{L}_G(\phi) = \mathbb{E}\big[\mathrm{MMD}_k^2\big(\pi_{\bar{\theta}}(\cdot \mid z_{g_\phi(z)}), \nu_z^*\big)\big]. \tag{16}$$

Here k is the fixed kernel specified in Appendix A.3, and the expectation is over contexts and target construction. We estimate the loss using policy samples generated with noise independent of target construction. Gradients reach ϕ through the budget input, with ν_z^* and $\bar{\theta}$ held fixed.

For fixed target-generating components, the expected loss is equivalent, up to an additive constant, to squared MMD against the averaged target $\mathbb{E}[\nu_z^* \mid z]$ (Appendix A.1). Exact matching need not be attainable, so target feasibility alone does not guarantee that the guided policy satisfies the predicted-cost constraint. At deployment, each action requires one guide evaluation and one flow evaluation; candidate generation, critic evaluation, and probability optimization are confined to training.

4.4 Practical Implementation

We train all components in a shared offline training loop, sampling a budget for each transition. The reference policy π_{ref} is trained throughout by MeanFlow regression on logged actions. During the first K_0 updates, the base policy π_θ uses the same objective, while the guide remains fixed. After warmup, the base policy learns from constrained action targets using Eq. 9, and the guide learns through action distribution matching using Eq. 16.

The critic ensembles for the base and guided policies are updated throughout training. The short-horizon cost predictors are trained on cumulative costs over H recorded steps, truncated at episode termination. The reward and cost critics use TD targets with continuation actions sampled from target copies of the corresponding base or guided policy. Each loss updates only its designated component, with training targets held fixed within each update. Algorithm 1 summarizes the procedure; component updates are described in Appendix A.4, and hyperparameters are listed in Table 3.

5 Experiments

Our experiments address three questions: (1) Can FIBA-Flow achieve competitive returns within budget without complete budget-compliant training trajectories? (2) How does removing low-cost trajectories affect performance, and can FIBA-Flow adapt to different budgets without retraining? (3) How do target construction and the guide affect return and budget compliance?

5.1 Experimental Setup

Benchmarks and data. We evaluate FIBA-Flow on the DSRL benchmark (Liu et al., 2023a), covering SafetyGym (Ji et al., 2023), BulletSafetyGym (Gronauer, 2022), and MetaDrive (Li et al.,

Table 1: DSRL results at budget $B = 10$. Each value is averaged over 20 evaluation episodes and 3 seeds. Gray: NC > 1; **bold**: NC ≤ 1; **blue**: Safe agents with the highest reward.

Task	CPQ		CDT		FISOR		TREBI		GAS		CAPS		CCAC		FIBA-Flow	
	NR↑	NC	NR↑	NC	NR↑	NC	NR↑	NC	NR↑	NC	NR↑	NC	NR↑	NC	NR↑	NC
PointButton1	0.71	11.17	0.49	8.95	0.10	1.96	**-0.04**	**0.79**	0.45	8.99	0.04	1.80	-0.02	6.90	**0.01**	**0.97**
PointButton2	0.41	17.14	0.44	8.79	0.10	2.49	**-0.02**	**0.92**	0.47	11.17	0.04	2.39	0.00	8.05	**0.03**	**0.96**
PointCircle1	**0.47**	**0.03**	**0.51**	**0.45**	0.51	14.80	**-0.07**	**0.00**	0.54	1.54	0.45	1.09	**0.45**	**0.16**	**0.52**	**0.06**
PointCircle2	-0.31	23.42	0.59	1.38	0.72	15.39	-0.08	10.26	0.60	5.39	**0.48**	**0.60**	**0.19**	**0.13**	**0.58**	**0.66**
PointGoal1	0.55	1.51	0.65	3.94	0.64	4.38	**0.09**	**0.40**	0.68	3.66	0.23	1.26	0.76	3.77	**0.26**	**0.86**
PointGoal2	0.38	3.94	0.61	7.42	**0.13**	**0.58**	**-0.04**	**0.33**	0.66	9.62	0.14	1.26	-0.01	3.45	**0.24**	**0.62**
PointPush1	0.21	10.06	0.24	2.79	0.26	2.25	**0.02**	**0.84**	0.28	3.77	0.14	2.08	**0.00**	**0.57**	**0.13**	**0.61**
PointPush2	0.15	2.70	0.21	4.00	0.25	2.52	**0.09**	**0.63**	0.23	5.02	0.15	3.72	-0.01	2.18	**0.06**	**0.49**
CarButton1	0.33	43.43	0.30	10.85	**-0.07**	**0.69**	**-0.02**	**0.83**	0.34	15.03	0.07	5.89	**0.00**	**0.00**	**-0.06**	**0.46**
CarButton2	0.05	23.01	0.17	9.31	**0.02**	**0.68**	-0.01	2.23	0.30	14.46	0.00	3.12	-0.60	8.87	**-0.27**	**0.46**
CarCircle1	-0.60	29.84	0.56	8.37	0.70	11.83	**-0.25**	**0.00**	0.72	17.64	0.10	2.25	-0.49	43.02	**0.39**	**0.40**
CarCircle2	-0.06	39.78	0.59	10.45	0.61	14.56	-0.15	10.22	0.66	18.12	**0.05**	**0.00**	**-0.24**	**0.00**	**0.38**	**0.13**
CarGoal1	0.52	9.38	0.55	4.44	0.42	1.94	**0.08**	**0.74**	0.57	3.83	0.22	1.51	**0.00**	**0.00**	**0.17**	**0.58**
CarGoal2	0.50	26.54	0.40	5.15	**0.04**	**0.45**	**0.07**	**0.87**	0.55	8.70	0.23	3.49	**0.00**	**0.00**	**-0.04**	**0.83**
CarPush1	0.02	15.41	0.31	2.76	**0.27**	**0.75**	**0.10**	**0.94**	0.30	2.06	0.17	1.53	**-1.67**	**0.22**	**0.19**	**0.75**
CarPush2	0.00	9.26	0.17	5.95	**0.14**	**0.89**	**-0.05**	**0.12**	0.21	6.43	0.08	3.77	**0.00**	**0.00**	**0.03**	**0.69**
SwimmerVel	-0.08	1.35	0.64	1.39	**-0.02**	**0.66**	0.05	21.94	0.66	1.22	**0.11**	**0.02**	**-0.08**	**0.02**	**0.47**	**0.01**
HopperVel	**-0.02**	**0.00**	0.74	2.46	0.26	4.44	0.04	2.20	0.79	3.01	**0.11**	**0.46**	**-0.02**	**0.00**	**0.72**	**0.55**
HalfCheetahVel	0.30	7.38	**0.97**	**0.91**	**0.88**	**0.00**	**0.05**	**0.01**	0.94	3.55	**0.89**	**0.27**	**0.89**	**0.73**	**0.95**	**0.31**
Walker2dVel	0.04	1.89	**0.80**	**0.66**	0.29	1.76	0.10	1.05	0.75	2.64	0.72	1.85	**0.50**	**0.70**	**0.61**	**0.94**
AntVel	**-1.01**	**0.00**	0.98	1.05	**0.87**	**0.01**	**0.38**	**0.53**	0.98	8.23	**0.90**	**0.11**	**-0.13**	**0.15**	**0.97**	**0.59**
SafetyGym Avg.	0.12	13.20	0.52	4.83	0.34	3.95	0.02	2.66	0.56	7.34	0.25	1.83	-0.02	3.76	**0.30**	**0.57**
BallRun	0.16	2.20	0.33	6.66	**0.15**	**0.00**	0.32	3.13	1.07	9.40	**0.14**	**0.00**	**0.33**	**0.60**	**0.21**	**0.56**
CarRun	**0.91**	**0.10**	**0.99**	**0.84**	**0.77**	**0.00**	0.94	1.11	-0.35	18.12	0.94	1.06	**0.95**	**0.33**	**0.98**	**0.67**
DroneRun	0.23	2.81	0.58	1.22	0.43	3.17	0.19	3.31	0.60	3.85	0.45	3.25	**0.52**	**0.21**	**0.37**	**0.00**
AntRun	**0.02**	**0.00**	0.71	2.03	0.48	1.43	0.47	1.80	0.70	2.33	**0.46**	**0.83**	**0.12**	**0.00**	**0.54**	**0.73**
BallCircle	**0.71**	**0.17**	0.73	1.60	**0.31**	**0.00**	**0.34**	**0.69**	0.00	15.87	**0.49**	**0.19**	**0.74**	**0.01**	**0.68**	**0.77**
CarCircle	**0.71**	**0.00**	0.72	2.02	**0.38**	**0.57**	0.02	1.85	0.63	1.38	**0.58**	**0.26**	**0.71**	**0.93**	**0.73**	**0.34**
DroneCircle	-0.20	1.95	0.56	1.42	**0.37**	**0.00**	0.17	1.80	**-0.26**	**0.32**	**0.38**	**0.01**	**0.24**	**0.47**	**0.36**	**0.57**
AntCircle	**0.00**	**0.00**	0.48	4.48	**0.16**	**0.00**	**0.06**	**0.67**	0.44	3.72	**0.31**	**0.00**	**0.00**	**0.00**	**0.34**	**0.18**
BulletGym Avg.	**0.32**	**0.90**	0.64	2.53	**0.38**	**0.65**	0.31	1.80	0.35	6.87	**0.47**	**0.70**	**0.45**	**0.32**	**0.53**	**0.48**
easysparse	**-0.06**	**0.16**	**0.00**	**0.01**	**0.10**	**0.38**	0.25	3.02	0.57	2.94	**0.08**	**0.49**	**-0.06**	**0.00**	**0.73**	**0.98**
easymean	**-0.07**	**0.10**	**0.19**	**0.24**	0.77	2.05	0.31	3.70	0.52	2.21	**0.31**	**0.69**	**-0.06**	**0.00**	**0.71**	**0.86**
easydense	**-0.06**	**0.16**	0.65	5.60	**0.25**	**0.70**	0.29	3.37	0.39	1.24	0.18	1.91	**-0.05**	**0.00**	**0.69**	**0.92**
mediumsparse	**-0.09**	**0.10**	0.95	2.49	0.86	2.39	**-0.02**	**0.34**	**0.23**	**0.49**	0.93	2.40	**-0.06**	**0.00**	**0.95**	**0.70**
mediummean	**-0.08**	**0.10**	**0.17**	**0.44**	0.92	2.93	**0.01**	**0.45**	**0.23**	**0.43**	0.92	3.43	**-0.06**	**0.00**	**0.94**	**0.75**
mediumdense	**-0.05**	**0.00**	0.80	4.16	0.80	1.77	**0.07**	**0.74**	**0.21**	**0.46**	0.93	2.17	**-0.05**	**0.00**	**0.93**	**0.74**
hardsparse	**-0.04**	**0.16**	0.46	3.48	0.52	2.53	0.09	1.49	**0.21**	**0.75**	**0.04**	**0.06**	**-0.04**	**0.00**	**0.54**	**0.83**
hardmean	**-0.05**	**0.10**	0.44	1.16	0.46	2.25	**0.00**	**0.51**	0.44	1.59	0.50	2.68	**-0.04**	**0.00**	**0.52**	**0.83**
harddense	**-0.02**	**0.24**	0.44	3.29	0.53	2.01	**0.00**	**0.31**	0.35	3.51	0.49	3.43	**-0.04**	**0.00**	**0.50**	**0.81**
MetaDrive Avg.	**-0.06**	**0.13**	0.45	2.32	0.58	1.89	0.11	1.55	0.35	1.51	0.49	1.92	**-0.05**	**0.00**	**0.72**	**0.83**

2022) environments. For the main comparison, we remove all complete training trajectories with cumulative cost $C(\tau) = \sum_t c_t \leq 10$ and evaluate at budget $B = 10$. All methods receive the same filtered training split before method-specific preprocessing.

Baselines. We compare against seven baselines. CPQ (Xu et al., 2022) inflates cost estimates for out-of-distribution actions and restricts reward backups to actions deemed safe. CDT (Liu et al., 2023b) predicts actions using a Transformer conditioned on reward and cost. FISOR (Zheng et al., 2024) learns a diffusion policy to maximize reward in feasible regions and reduce risk elsewhere. TREBI (Lin et al., 2023) plans trajectories through reward- and cost-guided diffusion under the remaining budget. GAS (Liu et al., 2026) combines transition-level return relabeling and augmentation with learned reward and cost goal functions for trajectory stitching. CAPS (Chemingui et al., 2025) switches among policies using reward and cost estimates and the remaining budget. CCAC (Guo et al., 2025b) regularizes actor–critic learning through constraint-conditioned data generation and out-of-distribution detection. We use the public MLP variant of GAS, and the IQL variant of CAPS.

Evaluation protocol. We report normalized return (NR) following the DSRL protocol and normalized cost (NC), defined as episode cost divided by the evaluation budget B. A method is considered budget-compliant on a task if NC ≤ 1.

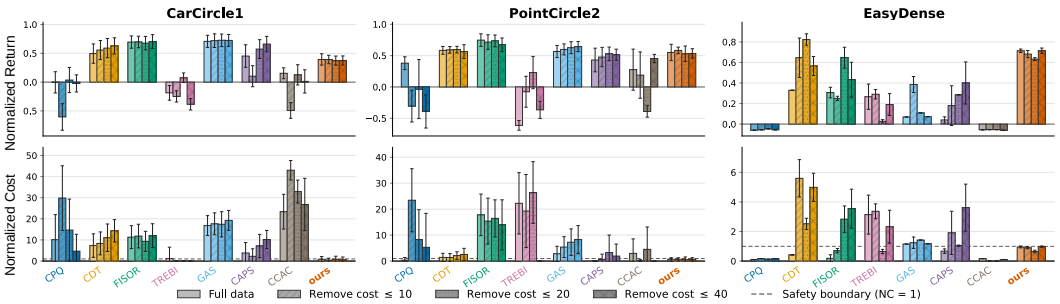

Figure 3: Effect of training-data filtering on three tasks at $B = 10$. We compare retaining all trajectories with removing those whose cumulative cost is at most 10, 20, or 40.

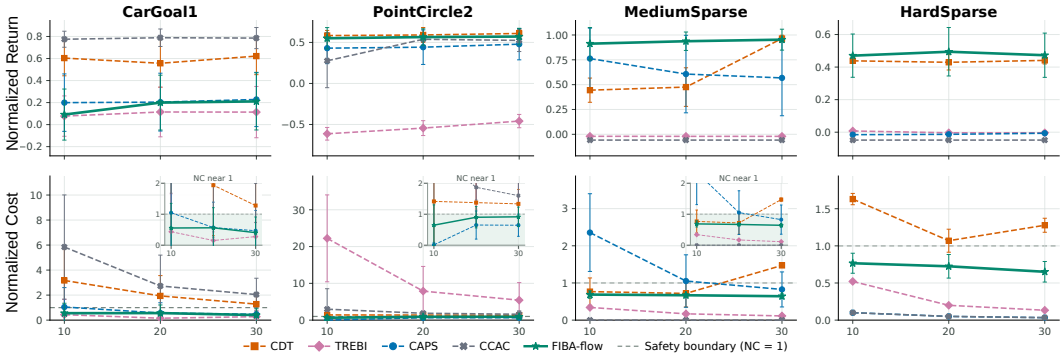

Figure 4: Zero-shot budget adaptation across four tasks without retraining, using datasets with all trajectories retained. Error bars indicate episode-level standard deviations.

5.2 Main Results

Table 1 shows that FIBA-Flow satisfies the cost constraint on all 38 tasks despite the absence of complete budget-compliant trajectories in the training data. It achieves the highest NR among budget-compliant methods on 25 tasks. The task-averaged NR/NC values are $0.30/0.57$ on SafetyGym, $0.72/0.83$ on MetaDrive, and $0.53/0.48$ on BulletSafetyGym. The return advantage is particularly pronounced in MetaDrive, where FIBA-Flow achieves the highest NR among budget-compliant methods on all nine tasks and the highest suite-level NR across all evaluated methods. These results support the feasibility of learning competitive, budget-compliant policies without complete budget-compliant training trajectories. Appendix Table 4 compares methods trained on the original datasets with all trajectories retained, reporting NR and NC averaged across three evaluation budgets. At deployment, FIBA-Flow requires one guide evaluation and one velocity-network evaluation per action. FIBA-Flow uses 1.19 MFLOPs per action, compared with CAPS (3.24), GAS (35.45), CDT (50.16), and FISOR (276.57), with a larger reduction relative to TREBI (272.88 GFLOPs).

5.3 Ablation Study

Effect of training-data filtering. We progressively remove low-cost trajectories from the training set while keeping the evaluation budget fixed at $B = 10$. Figure 3 shows that stronger filtering does not consistently improve safety or return, and its effects vary across methods and tasks. FIBA-Flow remains below the normalized-cost threshold across all four data settings on the three tasks, although its return also varies with filtering. These results highlight the importance of training-data composition beyond the evaluation budget.

Zero-shot budget adaptation. We evaluate four representative tasks using the full dataset. Figure 4 reports normalized return and cost with episode-level standard deviations. FIBA-Flow stays within budget at all three levels and achieves the highest return among methods feasible at every level on all four tasks. This demonstrates zero-shot budget adaptation without retraining.

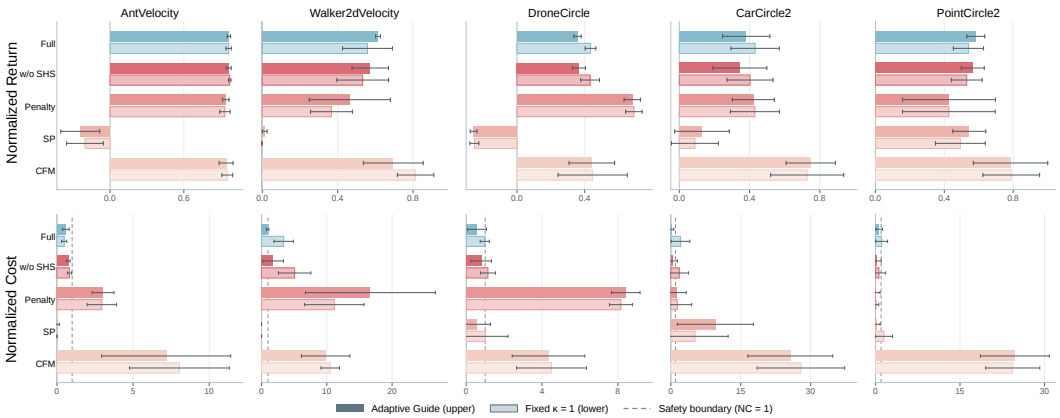

Figure 5: Mechanism ablations on five tasks. Each pair shows adaptive Guide and fixed $\kappa = 1$.

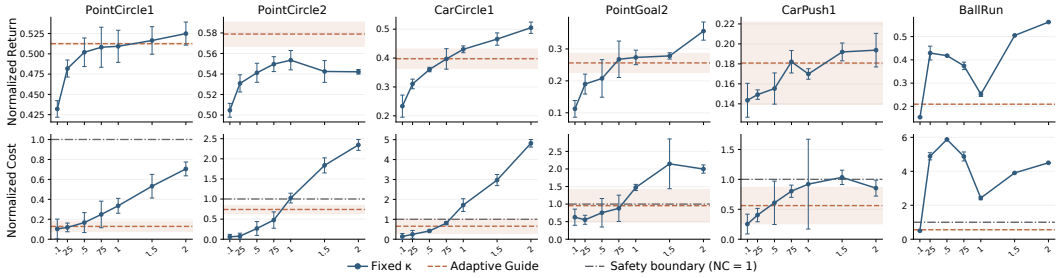

Figure 6: Sensitivity to fixed budget scale κ at $B = 10$. Guide achieves episode cost below the budget on all six tasks without task-specific manual tuning. Error bars and shaded bands indicate one standard deviation across three evaluation groups.

Component ablation. Figure 5 examines four variants of our target construction under fixed $\kappa = 1$ and adaptive Guide. *Conditional FM* (CFM) directly fits logged actions conditioned on the budget, replacing the target action distributions μ_z^*. *Self-proposal* (SP) draws candidate actions from the base policy π_θ rather than the reference policy π_{ref} fitted to the offline data, while retaining the logged action. *Penalty* replaces the explicit budget constraint with reward–cost weighting using a fixed coefficient $\lambda = 1$. Finally, *w/o short-horizon scoring* (w/o SHS) relies solely on the TD cost-to-go estimate. On these five tasks, CFM consistently violates the budget, SP generally yields low returns, and Penalty exhibits inconsistent budget compliance. Full with adaptive Guide satisfies the cost constraint on all five tasks, although it does not always achieve the highest return. Short-horizon scoring and adaptive conditioning each reduce cost on most tasks.

Adaptive guide. We evaluate the learned Guide against seven fixed values of κ (Figure 6), using the same selected checkpoint for all execution settings within each task. Different fixed budget scales produce different trade-offs between return and cost across tasks: increasing κ can improve return but may also cause budget violations. Although the guide does not outperform every fixed value on every task, it achieves normalized cost below one on all six evaluated tasks, without manually selecting a fixed κ for each task. These results support Guide as an automatic execution-time budget adaptation mechanism, rather than a uniformly optimal choice of budget scale.

6 Conclusion

We introduced FIBA-Flow for offline safe RL without requiring complete budget-compliant training trajectories. It learns a one-step policy from failure-informed, budget-constrained target action distributions. The guide learns to select budget scales whose action distributions approximate targets constructed using critics for the guided policy. After removing complete budget-compliant training trajectories, FIBA-Flow meets the cost constraint on all 38 DSRL tasks and achieves the highest normalized return among evaluated budget-compliant methods on 25 tasks.

AI USE STATEMENT

Generative AI tools were used to improve the language and readability of the manuscript. The authors take responsibility for the final text, scientific claims, and reported results.

ETHICS STATEMENT

This study uses existing offline datasets from simulated SafetyGym, BulletSafetyGym, and MetaDrive tasks. It does not involve human participants or personal data. Meeting cost constraints in these benchmarks does not establish safety for use in physical systems.

REPRODUCIBILITY STATEMENT

The main text describes the method, datasets, filtering rule, and evaluation protocol. The appendix provides the training and execution algorithms, hyperparameters, and implementation details. The random seeds used for training and evaluation are recorded in the experiment logs. We plan to provide videos of policy rollouts during inference as supplementary material.

REFERENCES

Iskander Azangulov, Peter Potaptchik, Qinyu Li, Eddie Aamari, George Deligiannidis, and Judith Rousseau. Adaptive diffusion guidance via stochastic optimal control. *arXiv preprint arXiv:2505.19367*, 2025.

Janaka Brahmanage and Akshat Kumar. Beyond hard constraints: Budget-conditioned reachability for safe offline reinforcement learning. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 36, pp. 581–590, 2026.

Returaj Burnwal, Nirav Pravinbhai Bhatt, and Balaraman Ravindran. Safemil: Learning offline safe imitation policy from non-preferred trajectories. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 19790–19798, 2026.

Chenyang Cao, Yucheng Xin, Silang Wu, Longxiang He, Zichen Yan, Junbo Tan, and Xueqian Wang. Fosp: Fine-tuning offline safe policy through world models. In *International Conference on Learning Representations*, volume 2025, pp. 39154–39185, 2025.

Yassine Chemingui, Aryan Deshwal, Honghao Wei, Alan Fern, and Jana Doppa. Constraint-adaptive policy switching for offline safe reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 15722–15730, 2025.

Yassine Chemingui, Aryan Deshwal, Alan Fern, Thanh Nguyen-Tang, and Jana Doppa. Online optimization for offline safe reinforcement learning. *Advances in Neural Information Processing Systems*, 38:140670–140698, 2026.

Rufeng Chen, Zhaofan Zhang, Zhejiang Yang, Hechang Chen, and Sihong Xie. Decoupled guidance diffusion for adaptive offline safe reinforcement learning, 2026. URL https://arxiv.org/abs/2605.02777.

Nicolas Espinosa-Dice, Yiyi Zhang, Yiding Chen, Bradley Guo, Owen Oertell, Gokul Swamy, Kiante Brantley, and Wen Sun. Scaling offline rl via efficient and expressive shortcut models. *Advances in Neural Information Processing Systems*, 38:2376–2414, 2026.

Alexandre Galashov, Ashwini Pokle, Arnaud Doucet, Arthur Gretton, Mauricio Delbracio, and Valentin De Bortoli. Learn to guide your diffusion model. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 129003–129039, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/d199e76b714c2611051e1b9e4bd882f1-Paper-Conference.pdf.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. *Advances in Neural Information Processing Systems*, 38:75460–75482, 2026.

Ze Gong, Akshat Kumar, and Pradeep Varakantham. Offline safe reinforcement learning using trajectory classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 16880–16887, 2025.

Sven Gronauer. Bullet-safety-gym: A framework for constrained reinforcement learning. Technical report, mediaTUM, 2022.

Junyu Guo, Zhi Zheng, Donghao Ying, Ming Jin, Shangding Gu, Costas Spanos, and Javad Lavaei. Don't trade off safety: Diffusion regularization for constrained offline rl. *arXiv preprint arXiv:2502.12391*, 2025a.

Zijian Guo, Weichao Zhou, Shengao Wang, and Wenchao Li. Constraint-conditioned actor-critic for offline safe reinforcement learning. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 41581–41602, 2025b. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/67089958e98b243d5cc1881ad60418b8-Paper-Conference.pdf.

Jiaming Ji, Borong Zhang, Jiayi Zhou, Xuehai Pan, Weidong Huang, Ruiyang Sun, Yiran Geng, Yifan Zhong, Josef Dai, and Yaodong Yang. Safety gymnasium: A unified safe reinforcement learning benchmark. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=WZmlxIuIGR.

Donghyeon Ki, JunHyeok Oh, Seong-Woong Shim, and Byung-Jun Lee. Prior-guided diffusion planning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 38:59790–59820, 2026.

Mingyu Kim, Young-Heon Kim, and Mijung Park. Safety-guided flow (sgf): A unified framework for negative guidance in safe generation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=EA80Zib9UI.

Prajwal Koirala, Zhanhong Jiang, Soumik Sarkar, and Cody Fleming. Latent safety-constrained policy approach for safe offline reinforcement learning. In *International Conference on Learning Representations*, volume 2025, pp. 61341–61367, 2025.

Jongmin Lee, Cosmin Paduraru, Daniel J. Mankowitz, Nicolas Heess, Doina Precup, Kee-Eung Kim, and Arthur Guez. Coptidice: Offline constrained reinforcement learning via stationary distribution correction estimation. In *The Tenth International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=FLA55mBee6Q.

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

Quanyi Li, Zhenghao Peng, Lan Feng, Qihang Zhang, Zhenghai Xue, and Bolei Zhou. Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

Qian Lin, Bo Tang, Zifan Wu, Chao Yu, Shangqin Mao, Qianlong Xie, Xingxing Wang, and Dong Wang. Safe offline reinforcement learning with real-time budget constraints. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 21127–21152. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/lin23h.html.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Zifan Liu, Xinran Li, and Jun Zhang. C2iql: Constraint-conditioned implicit q-learning for safe offline reinforcement learning. In *International Conference on Machine Learning*, pp. 38827–38841. PMLR, 2025.

Zifan Liu, Xinran Li, Shibo Chen, and Jun Zhang. Gas: Enhancing reward-cost balance of generative model-assisted offline safe rl. *arXiv preprint arXiv:2602.05323*, 2026.

Zuxin Liu, Zijian Guo, Haohong Lin, Yihang Yao, Jiacheng Zhu, Zhepeng Cen, Hanjiang Hu, Wenhao Yu, Tingnan Zhang, Jie Tan, et al. Datasets and benchmarks for offline safe reinforcement learning. *arXiv preprint arXiv:2306.09303*, 2023a.

Zuxin Liu, Zijian Guo, Yihang Yao, Zhepeng Cen, Wenhao Yu, Tingnan Zhang, and Ding Zhao. Constrained decision transformer for offline safe reinforcement learning. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 21611–21630. PMLR, 23–29 Jul 2023b. URL https://proceedings.mlr.press/v202/liu23m.html.

Franki Nguimatsia Tiofack, Théotime Le Hellard, Fabian Schramm, Nicolas Perrin-Gilbert, and Justin Carpentier. Guided flow policy: Learning from high-value actions in offline reinforcement learning. In *International Conference on Learning Representations*, volume 2026, pp. 30601–30625, 2026.

Seohong Park, Qiyang Li, and Sergey Levine. Flow q-learning. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 48104–48127. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/park25f.html.

Huikang Su, Dengyun Peng, Zifeng Zhuang, Yuhan Liu, Qiguang Chen, Donglin Wang, and Qinghe Liu. Boundary-to-region supervision for offline safe reinforcement learning. *Advances in Neural Information Processing Systems*, 38:57917–57946, 2026.

Mumuksh Tayal, Manan Tayal, and Ravi Prakash. Safe flow q-learning: Offline safe reinforcement learning with reachability-based flow policies. *arXiv preprint arXiv:2603.15136*, 2026.

Akifumi Wachi, Kohei Miyaguchi, Takumi Tanabe, Rei Sato, and Youhei Akimoto. A provable approach for end-to-end safe reinforcement learning. *Advances in Neural Information Processing Systems*, 38:77338–77372, 2026.

Zeyuan Wang, Da Li, Yulin Chen, Ye Shi, Liang Bai, Tianyuan Yu, and Yanwei Fu. One-step generative policies with q-learning: A reformulation of meanflow. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 26751–26759, 2026.

Zhendong Wang, Jonathan J. Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=AHvFDPi-FA.

Haoran Xu, Xianyuan Zhan, and Xiangyu Zhu. Constraints penalized q-learning for safe offline reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 8753–8760, 2022.

Ruiqi Xue, Ziqian Zhang, Lihe Li, Cong Guan, Lei Yuan, and Yang Yu. Adaptable safe policy learning from multi-task data with constraint prioritized decision transformer. *Advances in Neural Information Processing Systems*, 38:64765–64803, 2026.

Yihang Yao, Zhepeng Cen, Wenhao Ding, Haohong Lin, Shiqi Liu, Tingnan Zhang, Wenhao Yu, and Ding Zhao. Oasis: Conditional distribution shaping for offline safe reinforcement learning. *Advances in Neural Information Processing Systems*, 37:78451–78478, 2024.

Guojian Zhan, Letian Tao, Pengcheng Wang, Yixiao Wang, Yuxin Chen, Yiheng Li, Hongyang Li, Masayoshi Tomizuka, and Shengbo Li. Mean flow policy with instantaneous velocity constraint for one-step action generation. In *International Conference on Learning Representations*, volume 2026, pp. 157132–157154, 2026.

He Zhang, Ying Sun, and Hui Xiong. Goldenstart: Q-guided priors and entropy control for distilling flow policies. *arXiv preprint arXiv:2603.14245*, 2026.

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.

Yinan Zheng, Jianxiong Li, Dongjie Yu, Yujie Yang, Shengbo Li, Xianyuan Zhan, and Jingjing Liu. Safe offline reinforcement learning with feasibility-guided diffusion model. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 3838–3867, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/10a3b1c30b8cceb507b9e8ddcc9a1a6a-Paper-Conference.pdf.

A APPENDIX

A.1 THEORETICAL ANALYSIS

All results below fix the context z and the components used to construct the training targets.

Budget-constrained target probabilities. For finite candidate scores and $\eta > 0$, Eq. 5 is feasible exactly when $[\beta]_+ \geq \min_i \widehat{D}_i$. Its objective is continuous and strictly concave on the simplex, so its optimizer is unique whenever the feasible set is nonempty. For $[\beta]_+ > \min_i \widehat{D}_i$, introducing a cost multiplier $\lambda \geq 0$ gives the stationarity condition

$$\widehat{R}_i - \lambda \widehat{D}_i - \eta\big(\log(N w_i) + 1\big) = \text{constant in } i. \tag{17}$$

Normalizing these weights gives Eq. 6. Differentiation yields

$$\frac{\mathrm{d}w_i(\lambda)}{\mathrm{d}\lambda} = -\frac{w_i(\lambda)}{\eta}\left(\widehat{D}_i - \sum_k w_k(\lambda)\widehat{D}_k\right), \tag{18}$$

and hence the variance identity in Eq. 6. The mean score decreases continuously to $\min_i \widehat{D}_i$ as $\lambda \to \infty$, strictly unless all scores coincide. Thus $\lambda^* = 0$ when $w(0)$ is feasible; otherwise, the unique positive crossing of $[\beta]_+$ gives λ^*. The resulting weights satisfy the KKT conditions and are optimal. At $[\beta]_+ = \min_i \widehat{D}_i$, the optimizer is supported on minimum-score actions, with weights proportional to $\exp(\widehat{R}_i/\eta)$ there. Below this minimum, the same weights define the fallback, whose mean score is $\min_i \widehat{D}_i$ and remains above the constraint threshold $[\beta]_+$. The same derivation applies to Eq. 14.

Consistency of flow learning. We prove Proposition 1. Write $\bar{\mu}_z = \mathbb{E}[\mu_z^* \mid z]$. Sampling $a \mid \mu_z^*, z \sim \mu_z^*$ rewrites the weighted sum in Eq. 9 as an expectation over a target action. With fresh $\epsilon \sim \mathcal{N}(0, I)$, set $X_s = (1-s)a + s\epsilon$. For each realized target, the velocity used in the update is $v(x,s) = \mathbb{E}[\epsilon - a \mid X_s = x, \mu_z^*, z]$. Its marginal counterpart satisfies

$$\bar{v}(x,s) = \mathbb{E}[\epsilon - a \mid X_s = x, z] = \mathbb{E}[v(X_s,s) \mid X_s = x, z]. \tag{19}$$

We suppress z in $u(x,r,s) = u_\theta(z,x,r,s)$.

Conditions. The target is a measurable random probability measure with finite expected second moment, and the update residual is square-integrable. Gaussian noise is independent of the target and sampled action; (r,s) is independent of all three and has a density positive almost everywhere on $0 < r < s < 1$. On every interval $[\delta, 1]$, $\delta > 0$, $\bar{v}$ is continuous, continuously differentiable in x, locally Lipschitz in x, and has at most linear growth, uniformly in time. The functions $u, \partial_s u, \nabla_x u$ are continuous in the interior, and u extends finitely and continuously to $0 < r \leq s \leq 1$. The sampler uses a finite measurable endpoint value satisfying $u(x,r,1) \to u(x,0,1)$ as $r \downarrow 0$ for Gaussian-almost every x.

Proof. The sampled action has marginal law $\bar{\mu}_z$. Differentiating $\mathbb{E}[f(X_s) \mid z]$ for smooth, compactly supported f gives $\frac{\mathrm{d}}{\mathrm{d}s}\mathbb{E}[f(X_s) \mid z] = \mathbb{E}[\nabla f(X_s) \cdot \bar{v}(X_s,s) \mid z]$. Thus $\bar{v}$ satisfies the continuity equation for the interpolation laws; the stated regularity gives a unique characteristic flow transporting these laws at positive times.

At a function-space stationary point of the stop-gradient update, the expected prediction–target residual is orthogonal to every smooth, compactly supported perturbation. Conditioning on (X_s, r, s) and using Eq. 19 therefore gives

$$u + (s-r)\big(\partial_s u + (\nabla_x u)\bar{v}\big) = \bar{v}. \tag{20}$$

The positive interpolation and time-sampling densities give this identity almost everywhere; continuity extends it throughout the interior. Along a characteristic $\dot{x}_s = \bar{v}(x_s, s)$ starting from $x_1 = \epsilon$, the chain rule gives

$$\frac{\mathrm{d}}{\mathrm{d}s}\big[(s-r)u(x_s,r,s)\big] = \dot{x}_s. \tag{21}$$

Table 2: Core notation used in FIBA-Flow. Sampling counts and regularization parameters are listed in Table 3.

Notation	Meaning
Context and policies	
$z = (h, T-t, \beta, B)$	History, remaining steps, signed remaining budget, and total episode budget.
π_{ref}	Reference policy fitted to logged actions.
$u_\theta,\ \pi_\theta$	Average-velocity network and the base policy it defines through one-step sampling.
$g_\phi,\ \kappa$	Budget guide and its selected internal budget scale.
$z_\kappa,\ \pi$	Adjusted flow context and guided policy $\pi(\cdot \mid z) = \pi_\theta(\cdot \mid z_{g_\phi(z)})$.
Base-policy targets	
$\widehat{R}_i,\ \widehat{L}_i$	Predicted return and cost after candidate action a_i, with base-policy continuation.
$\widehat{M}_i$	Predicted cost over H recorded steps, truncated at episode termination.
$\widehat{D}_i$	Candidate cost score combining TD and short-window predictions.
$w^*,\ \mu_z^*$	Candidate-action probabilities and the resulting target action distribution.
Guide targets	
$\widehat{Q}_R^\pi,\ \widehat{Q}_C^\pi$	Predicted return and cost with subsequent actions selected by the guided policy.
$\bar{R}_j,\ \bar{C}_j,\ \widetilde{R}_j$	Mean predicted return and cost for candidate condition κ_j, and its normalized return score.
$p^*,\ \nu_z^*$	Candidate-condition probabilities and the resulting empirical action mixture.

Integrating from r to 1 and using the continuous diagonal extension yields $\epsilon-(1-r)u(\epsilon,r,1) = x_r$. Since x_r has the law of X_r, the endpoint condition and $X_r \to a$ in L^2 imply

$$\pi_\theta(\cdot \mid z) = \mathrm{Law}\big(\epsilon - u(\epsilon, 0, 1)\big) = \bar{\mu}_z. \tag{22}$$

This proves Proposition 1.

For the same fixed TD-cost evaluator $\widehat{Q}_C^{\pi_\theta}(z, a_i) = \widehat{L}_i$, assume integrability under $\bar{\mu}_z$ and a finite expected fallback excess. Since $\widehat{L}_i \le \widehat{D}_i$, each realized target satisfies

$$\sum_i w_i^* \widehat{L}_i \le \sum_i w_i^* \widehat{D}_i \le [\beta]_+ + \big[\min_i \widehat{D}_i - [\beta]_+\big]_+. \tag{23}$$

Averaging over target construction and using the distribution identity above gives

$$\mathbb{E}_{a\sim\pi_\theta(\cdot|z)}[\widehat{Q}_C^{\pi_\theta}(z,a)] \le [\beta]_+ + \mathbb{E}\Big[\big[\min_i \widehat{D}_i - [\beta]_+\big]_+ \mid z\Big]. \tag{24}$$

These conclusions concern fixed-target consistency and predicted cost; they do not establish joint-training convergence or compliance with true environment costs.

Averaged guide target matching. Fix $\bar{\theta}$, the target-sampling law and the bounded continuous positive-definite kernel k, independently of the guide parameter ϕ being optimized. Write $\bar{\nu}_z = \mathbb{E}[\nu_z^* \mid z]$ and let $m(P) = \int k(a,\cdot)P(\mathrm{d}a)$ denote the kernel mean embedding. Linearity gives $\mathbb{E}[m(\nu_z^*) \mid z] = m(\bar{\nu}_z)$. Expanding the squared embedding distance therefore yields

$$\begin{aligned}&\mathbb{E}\big[\mathrm{MMD}_k^2\big(\pi_{\bar{\theta}}(\cdot \mid z_{g_\phi(z)}), \nu_z^*\big) \mid z\big] \\ &\quad = \mathrm{MMD}_k^2\big(\pi_{\bar{\theta}}(\cdot \mid z_{g_\phi(z)}), \bar{\nu}_z\big) + \mathbb{E}\big[\mathrm{MMD}_k^2(\nu_z^*, \bar{\nu}_z) \mid z\big].\end{aligned} \tag{25}$$

The cross term vanishes, and the final term is independent of ϕ. Consequently, the expected guide objective equals squared MMD to the averaged target plus an additive constant. The guide's distribution family need not contain this target.

A.2 Notation

A.3 Implementation Details

Hyperparameters The EMA update assigns weight 0.005 to the current network and 0.995 to its previous target copy. The short-horizon window is fixed at $H = 16$; recorded cost sums stop at episode termination. Guide return scores and the equally weighted Gaussian-kernel mixture are defined in Eqs. 26 and 27, respectively.

Table 3: Default training and guide settings of FIBA-Flow.

Hyperparameter	Default value
Optimization and networks	
Optimizer	Adam
Learning rate	3×10^{-4}
Batch size	256
Warmup updates K_0	10,000
Hidden width (flow / critics / guide)	256 / 256 / 64
Target-network EMA update rate	0.005
Random seed	{2026,2027,2028}
Value estimation and action targets	
Reward discount γ	0.99
Reward used for critic training	$r/10$
Training budget range	$[1, 80]$
Candidate actions N	5 (4 generated + 1 logged)
Short-horizon cost window H	16 steps
Action-target regularization η	0.01
Budget guide	
Candidate scales ($J = 3$)	$\{0.1, 1, 2\}$
Continuous guide output range	$[0.1, 2]$
Target actions per candidate scale M_G	4
Actions sampled using the guide for MMD K_s	4
Guide-target regularization η_G	1
Return-score normalization floor	10^{-6}
Gaussian-kernel bandwidths	$\{0.1, 0.3, 1\}$

Guide return normalization. For each context, we normalize the predicted returns across the J candidate budget conditions:

$$\widetilde{R}_j = \frac{\bar{R}_j - \mathrm{mean}_\ell(\bar{R}_\ell)}{\max\{\mathrm{std}_\ell(\bar{R}_\ell), 10^{-6}\}}. \tag{26}$$

The mean and standard deviation are computed over all candidate conditions, with divisor J for the variance. The mean predicted costs $\bar{C}_j$ remain unnormalized and are expressed in the same units as the constraint threshold $[\beta]_+$.

MMD kernel. The guide loss uses an equally weighted mixture of Gaussian kernels:

$$k(a, a') = \frac{1}{3} \sum_{\sigma \in \{0.1, 0.3, 1\}} \exp\left(-\frac{\|a - a'\|_2^2}{2\sigma^2} \right). \tag{27}$$

The bandwidths remain fixed throughout training and are independent of the sampled actions and guide parameters.

A.4 Training and Execution

Algorithms 1 and 2 summarize training and deployment. All target networks are EMA copies; $\bar{\pi}_{\mathrm{ref}}$ denotes the target reference policy. MeanFlow fitting to a logged action uses Eq. 9 with that action as the sole target, of weight one, and the network being fitted in place of u_θ. Reward and total-cost critics use squared regression losses with policy-specific TD targets; reward returns use discount γ, whereas costs are undiscounted. Bootstrap terms are zero at termination.

Each update changes only its designated component. Candidate actions, scores, and probabilities used to construct targets are held fixed during policy and guide updates. We compute the target probabilities using Eq. 5 and 14 and apply the corresponding fallback when the constraint is infeasible. During a guide update, $\bar{\theta}$ is fixed, while gradients pass through the flow's conditioning input to ϕ.

Algorithm 1 FIBA-Flow: Offline Training

Require: Offline dataset $\mathcal{D}$; budget sampler on $[1, 80]$; horizon T; training updates K; settings in Table 3.
Ensure: Base policy π_θ and budget guide g_ϕ.

Initialize π_θ, g_ϕ, and both critic ensembles; initialize π_{ref} from π_θ and create target copies.
for $k = 1, \ldots, K$ **do**
  Sample an offline minibatch and a budget B for each transition.
  Set $\beta \leftarrow B - \sum_{\tau<t} c_\tau$ using the recorded episode; form $z = (h, T-t, \beta, B)$ and $z' = (h', T-t-1, \beta-c, B)$.
  Fit π_{ref} to logged actions by MeanFlow regression.
  Update the base critic ensemble using continuation actions $a' \sim \pi_{\bar\theta}(\cdot \mid z')$; fit short-window predictors to recorded cost sums.
  Update the guided critic ensemble using continuation actions $a' \sim \pi_{\bar\theta}(\cdot \mid z'_{g_{\bar\phi}(z')})$.
  if $k \le K_0$ **then**
    Fit π_θ to logged actions by MeanFlow regression; keep g_ϕ fixed.
  else
    Constrained action targets and flow update
    Draw $N-1$ actions from $\bar\pi_{\mathrm{ref}}(\cdot \mid z)$ and append the logged action.
    Compute $\widehat{R}_i$, $\widehat{L}_i$, and $\widehat{M}_i$; form $\widehat{D}_i$ using Eq. 4.
    Solve for w^* using Eq. 5 and form μ_z^* using Eq. 7.
    Draw Gaussian noise ϵ and a time pair (r, s), $0 < r < s < 1$, independently of each other and the candidate pool.
    Form $x_{i,s} = (1-s)a_i + s\epsilon$ for all i and compute the targets in Eq. 8.
    Update θ with the full weighted loss $\mathcal{L}_{\mathrm{flow}}$ in Eq. 9.
    Guide target and distribution matching
    Set $(\kappa_1, \kappa_2, \kappa_3) = (0.1, 1, 2)$.
    Draw M_G independent Gaussian noise samples ϵ_m, shared across candidate conditions.
    Generate $a_{jm} = \epsilon_m - u_{\bar\theta}(z_{\kappa_j}, \epsilon_m, 0, 1)$ for all j, m.
    Evaluate actions at the actual context z to obtain $\bar{R}_j, \bar{C}_j$ from Eq. 13; normalize returns using Eq. 26.
    Solve for p^* using Eq. 14 and form ν_z^* using Eq. 15.
    Draw K_s fresh, independent Gaussian noise samples $\widetilde{\epsilon}_b$ and generate actions using the budget scale selected by the guide: $\widetilde{a}_b = \widetilde{\epsilon}_b - u_{\bar\theta}(z_{g_\phi(z)}, \widetilde{\epsilon}_b, 0, 1)$.
    Update ϕ using a sample estimate of the loss in Eq. 16.
  end if
  Update all target networks by EMA.
end for
return π_θ, g_ϕ.

Algorithm 2 FIBA-Flow: Budget-Conditioned Execution

Require: Trained u_θ, g_ϕ; episode budget B; horizon T; initial observation o_0.

Initialize history h from o_0 and set $\beta \leftarrow B$.
for $t = 0, \ldots, T-1$ **do**
  Form the actual context $z = (h, T-t, \beta, B)$.
  Set $\kappa \leftarrow g_\phi(z)$ and form z_κ using Eq. 11.
  Draw $\epsilon \sim \mathcal{N}(0, I)$ and generate $a = \epsilon - u_\theta(z_\kappa, \epsilon, 0, 1)$.
  Execute a; observe the reward, cost c, and next observation.
  Set $\beta \leftarrow \beta - c$ and update h with the observed transition.
  if the episode terminates **then**
    break
  end if
end for

The actual budget B is fixed throughout the episode, and negative balances are retained. Each environment step requires one guide evaluation and one flow evaluation.

A.5 More results

Performance under a broader budget-selection rule. Table 4 reports FIBA-Flow under a deliberately more permissive operating rule than strict zero-shot budget transfer: for each task, a differ-

Table 4: Full-data DSRL results averaged equally over three budgets (SafetyGym: $20, 40, 80$; BulletGym and MetaDrive: $10, 20, 40$). Gray: NC > 1; **bold**: NC ≤ 1; **blue**: highest NR among methods with NC ≤ 1.

Task	CPQ		CDT		FISOR		TREBI		GAS		CAPS		CCAC		FIBA-Flow	
	NR↑	NC↓	NR↑	NC↓	NR↑	NC↓	NR↑	NC↓	NR↑	NC↓	NR↑	NC↓	NR↑	NC↓	NR↑	NC↓
PointButton1	0.69	3.20	0.53	1.68	**0.06**	**0.45**	**0.07**	**0.77**	0.46	3.69	**0.03**	**0.50**	0.70	4.30	**0.17**	**0.94**
PointButton2	0.58	4.30	0.46	1.57	**0.10**	**0.42**	0.16	1.35	0.42	2.69	**0.14**	**0.75**	0.63	4.81	**0.20**	**0.93**
PointCircle1	**0.43**	**0.75**	**0.59**	**0.69**	0.56	3.40	**0.23**	**0.02**	**0.60**	**0.90**	**0.50**	**0.78**	0.67	3.10	**0.58**	**0.83**
PointCircle2	0.24	3.58	0.64	1.05	0.72	4.79	-0.09	4.26	0.61	1.28	**0.51**	**0.80**	0.53	1.93	**0.60**	**0.79**
PointGoal1	**0.57**	**0.35**	0.69	1.12	0.64	1.09	**0.35**	**0.32**	0.70	1.27	**0.47**	**0.53**	0.36	2.53	**0.45**	**0.61**
PointGoal2	0.40	1.31	0.59	1.34	**0.30**	**0.78**	**0.25**	**0.29**	0.65	3.07	**0.33**	**0.80**	0.62	5.90	**0.35**	**0.97**
PointPush1	**0.20**	**0.83**	**0.24**	**0.48**	**0.34**	**0.51**	**0.13**	**0.34**	0.28	1.23	**0.19**	**0.29**	-0.12	1.61	**0.17**	**0.35**
PointPush2	0.11	1.04	**0.21**	**0.65**	**0.30**	**0.57**	**0.17**	**0.41**	0.26	1.12	**0.13**	**0.64**	0.15	1.41	**0.15**	**0.95**
CarButton1	0.42	9.66	0.21	1.60	**-0.23**	**0.01**	0.10	1.25	0.38	4.69	**-0.01**	**0.30**	0.44	10.72	**0.02**	**0.92**
CarButton2	0.37	12.51	0.13	1.58	**-0.02**	**0.15**	**0.00**	**0.59**	0.37	3.77	**-0.08**	**0.32**	0.53	8.73	**-0.10**	**0.41**
CarCircle1	0.02	2.29	0.60	1.73	0.71	3.18	**-0.05**	**0.29**	0.76	5.13	0.54	1.51	-0.24	10.79	**0.53**	**0.98**
CarCircle2	0.44	2.69	0.66	2.53	0.63	4.28	0.15	1.53	0.71	6.50	0.50	1.55	0.54	2.26	**0.49**	**0.54**
CarGoal1	0.79	1.42	0.66	1.21	**0.52**	**0.60**	**0.14**	**0.19**	0.65	1.25	**0.33**	**0.38**	0.80	1.61	**0.30**	**0.52**
CarGoal2	0.65	3.75	0.48	1.25	**0.04**	**0.15**	0.20	1.25	0.56	2.96	**0.16**	**0.62**	0.96	5.23	**0.08**	**0.52**
CarPush1	**-0.03**	**0.95**	**0.31**	**0.40**	**0.23**	**0.34**	**0.17**	**0.34**	**0.30**	**0.84**	**0.20**	**0.31**	**-0.63**	**0.62**	**0.19**	**0.63**
CarPush2	0.24	4.25	0.19	1.30	**0.16**	**0.14**	**0.00**	**0.10**	0.22	1.56	**0.07**	**0.75**	0.15	8.24	**0.06**	**0.39**
SwimmerVel	0.13	2.66	**0.66**	**0.96**	**-0.02**	**0.00**	0.10	3.97	**0.68**	**0.81**	0.43	1.58	**-0.09**	**0.25**	**0.52**	**0.63**
HopperVel	0.14	2.11	**0.63**	**0.61**	**0.19**	**0.20**	**0.07**	**0.37**	0.84	1.29	**0.41**	**0.70**	**0.05**	**0.65**	**0.71**	**0.56**
HalfCheetahVel	**0.29**	**0.74**	**1.00**	**0.01**	**0.89**	**0.00**	**0.15**	**0.00**	**0.94**	**0.54**	**0.94**	**0.77**	**0.89**	**0.98**	**0.99**	**0.93**
Walker2dVel	**0.04**	**0.21**	**0.78**	**0.06**	**0.50**	**0.47**	**0.09**	**0.09**	**0.80**	**0.19**	**0.80**	**0.62**	0.41	2.42	**0.78**	**0.93**
AntVel	**-1.01**	**0.00**	**0.98**	**0.39**	**0.89**	**0.00**	**0.23**	**0.07**	0.97	1.66	**0.95**	**0.64**	**-1.01**	**0.00**	**0.99**	**0.86**
SafetyGym Avg.	0.27	2.79	0.54	1.06	0.36	1.03	**0.12**	**0.85**	0.58	2.21	**0.36**	**0.72**	0.30	3.72	**0.39**	**0.72**
BallRun	0.22	1.27	0.39	1.16	**0.12**	**0.35**	**0.19**	**0.91**	0.36	3.64	**0.19**	**0.94**	**-0.12**	**0.02**	**0.47**	**0.91**
CarRun	0.95	1.79	**0.99**	**0.65**	**0.69**	**0.00**	**0.99**	**0.55**	-0.36	10.56	**0.97**	**0.25**	**-0.36**	**0.00**	**1.00**	**0.58**
DroneRun	0.33	3.52	**0.63**	**0.79**	**0.27**	**0.53**	0.50	2.85	**0.57**	**0.06**	0.47	2.19	**0.37**	**0.86**	**0.51**	**0.97**
AntRun	**0.03**	**0.02**	**0.72**	**0.91**	**0.41**	**0.20**	**0.55**	**0.85**	0.70	1.00	**0.61**	**0.90**	**0.06**	**0.00**	**0.58**	**1.00**
BallCircle	**0.64**	**0.76**	0.77	1.07	**0.33**	**0.00**	**0.37**	**0.36**	0.00	9.00	**0.69**	**0.59**	**0.80**	**0.45**	**0.76**	**0.96**
CarCircle	**0.71**	**0.33**	**0.75**	**0.95**	**0.45**	**0.00**	0.08	4.05	**0.67**	**0.97**	**0.69**	**0.65**	0.73	1.72	**0.75**	**0.78**
DroneCircle	-0.22	1.28	**0.63**	**0.98**	**0.47**	**0.00**	-0.15	1.05	**-0.26**	**0.00**	**0.55**	**0.67**	**0.38**	**0.61**	**0.33**	**0.27**
AntCircle	**0.00**	**0.00**	0.54	1.78	**0.26**	**0.00**	0.05	1.67	0.45	1.97	**0.41**	**0.15**	**0.61**	**0.48**	**0.42**	**0.71**
BulletGym Avg.	0.33	1.12	0.68	1.04	**0.37**	**0.13**	0.32	1.53	0.27	3.40	**0.57**	**0.79**	**0.31**	**0.52**	**0.60**	**0.77**
EasySparse	**-0.06**	**0.07**	**0.17**	**0.23**	**0.41**	**0.28**	**0.03**	**0.44**	**0.07**	**0.68**	**0.11**	**0.34**	**-0.06**	**0.10**	**0.82**	**0.99**
EasyMean	**-0.07**	**0.07**	**0.45**	**0.54**	**0.42**	**0.35**	**0.02**	**0.45**	0.44	1.04	**0.01**	**0.20**	**-0.07**	**0.06**	**0.82**	**0.98**
EasyDense	**-0.06**	**0.03**	**0.32**	**0.62**	**0.42**	**0.04**	0.31	1.95	**0.07**	**0.70**	**0.10**	**0.19**	**-0.06**	**0.10**	**0.80**	**0.96**
MediumSparse	**-0.08**	**0.07**	0.87	1.10	**0.37**	**0.04**	**-0.02**	**0.20**	**0.14**	**0.26**	**0.60**	**0.74**	**-0.06**	**0.00**	**0.97**	**0.68**
MediumMean	**-0.08**	**0.05**	**0.45**	**0.75**	**0.47**	**0.41**	**0.05**	**0.49**	**0.23**	**0.36**	**0.66**	**0.94**	**-0.08**	**0.09**	**0.95**	**0.69**
MediumDense	**-0.07**	**0.07**	0.88	2.41	**0.48**	**0.02**	0.25	1.32	**0.21**	**0.08**	**0.69**	**0.56**	**-0.05**	**0.00**	**0.95**	**0.65**
HardSparse	**-0.05**	**0.06**	**0.25**	**0.41**	**0.29**	**0.11**	**0.00**	**0.27**	0.22	1.07	**0.45**	**0.72**	**-0.05**	**0.06**	**0.55**	**0.73**
HardMean	**-0.05**	**0.06**	**0.33**	**0.97**	**0.27**	**0.03**	**0.00**	**0.30**	**0.20**	**0.60**	**0.28**	**0.25**	**-0.05**	**0.06**	**0.53**	**0.68**
HardDense	**-0.04**	**0.08**	**0.08**	**0.21**	**0.34**	**0.11**	0.08	1.06	0.27	1.62	**0.37**	**0.67**	**-0.04**	**0.00**	**0.53**	**0.71**
MetaDrive Avg.	**-0.06**	**0.06**	**0.42**	**0.80**	**0.38**	**0.15**	**0.08**	**0.72**	**0.20**	**0.71**	**0.36**	**0.51**	**-0.06**	**0.05**	**0.77**	**0.79**

ent full-data checkpoint may be selected for each budget, and feasibility is imposed on the equally weighted three-budget mean NC rather than on every budget separately.

This distinction matters because the rule exposes the return–cost frontier available to the method. It should therefore be read as an achievable operating envelope, not as evidence that one unchanged checkpoint is safe at every budget. FIBA-Flow attains the highest task-averaged NR among methods whose tabulated NC is at most one in all three suites. In SafetyGym it reaches NR/NC of $0.393/0.724$, compared with $0.36/0.72$ for the strongest safe baseline (CAPS); In BulletGym it reaches $0.600/0.772$, versus $0.57/0.79$ for CAPS. In MetaDrive it reaches $0.769/0.787$, versus $0.42/0.80$ for the strongest safe baseline by return (CDT). The SafetyGym and BulletGym gains are therefore obtained at nearly the same aggregate normalized cost as the strongest safe baseline, while the MetaDrive return gap is substantially larger with NC below one.

Sensitivity to reference candidate count. We vary $N - 1 \in \{0, 1, 2, 4, 8\}$, where each target is constructed from K actions generated by the reference policy and one logged action. Thus, at $K = 0$, the constrained target has no alternative action to which it can shift probability. All variants

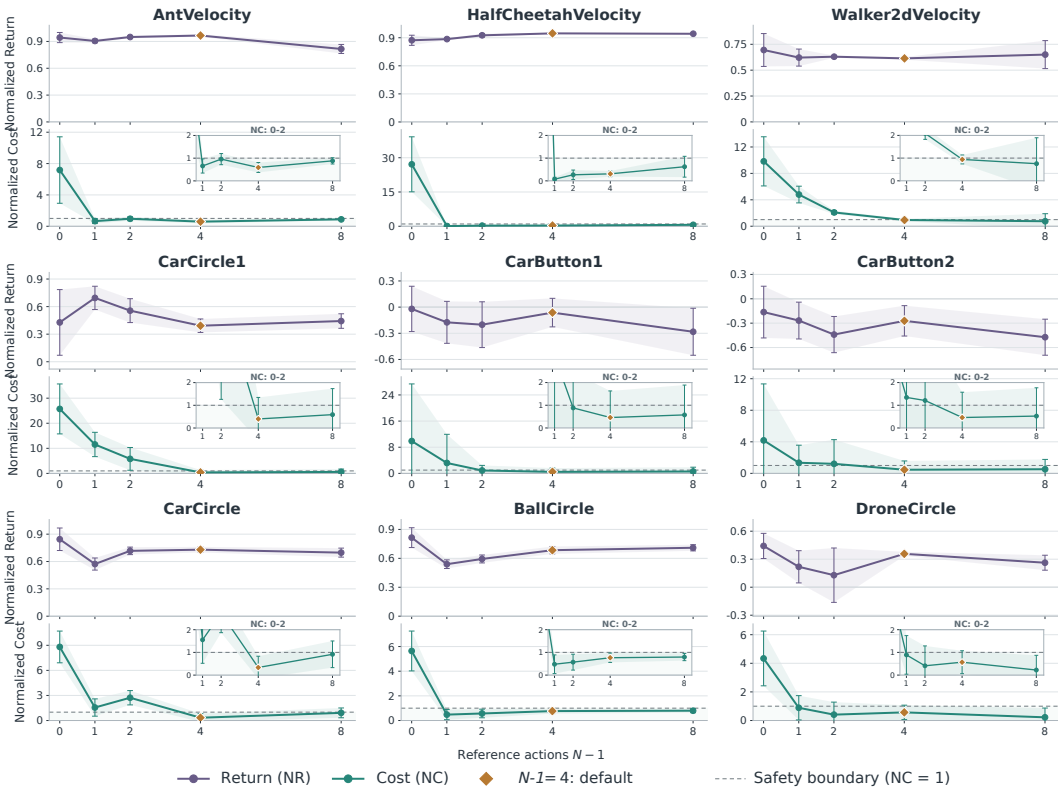

Figure 7: Sensitivity to the number of candidate actions generated by the reference policy.

use the main experiment's filtered training split, budget $B = 10$, short-horizon window $H = 16$, and the same task-specific number of training updates; the $K = 4$ points reuse the main-table results exactly. Figure 7 shows episode means and sample standard deviations. The nine displayed tasks are a post hoc subset of cases selected from the 38-task sweep for a transition from budget violation at $N - 1 = 0$ to compliance at $K = 4$, so the figure illustrates that transition rather than its prevalence across tasks. Among these nine tasks, 4 meet the normalized-cost limit at $N - 1 = 1$, 5 at $N - 1 = 2$, and all 9 at $N - 1 = 4$. This pattern is consistent with reference proposals giving the budget-constrained target lower-cost alternatives to the logged action. Increasing to $N - 1 = 8$ keeps all nine within budget, but raises mean normalized cost relative to $N - 1 = 4$ on seven tasks and does not systematically improve return. More candidates therefore need not yield monotonic gains, nor does this selected subset establish a universally optimal candidate count.

Sensitivity to short-horizon scoring length. We vary how many predicted short-horizon costs enter the scoring of reference actions, while keeping the training data, candidate count ($N - 1 = 4$), evaluation budget ($B = 10$), task-specific selected training step, and noise-free inference fixed. Every variant trains the same 16 short-cost prediction heads; changing H changes which predictions contribute to target construction, rather than model capacity. In Fig. 8, the default $H = 16$ satisfies the mean cost budget on all eight displayed tasks, compared with three tasks at $H = 0$. Across these tasks, mean NR/NC changes from $0.256/1.073$ at $H = 0$ to $0.326/0.555$ at $H = 16$. This pattern is consistent with short-horizon cost estimates helping distinguish reference actions by their predicted near-term risk. The curves are not monotonic, however, and $H = 16$ does not maximize return on every task. Since these eight tasks were selected after a preliminary inspection, their averages describe illustrative cases rather than an estimated effect across all 38 tasks.

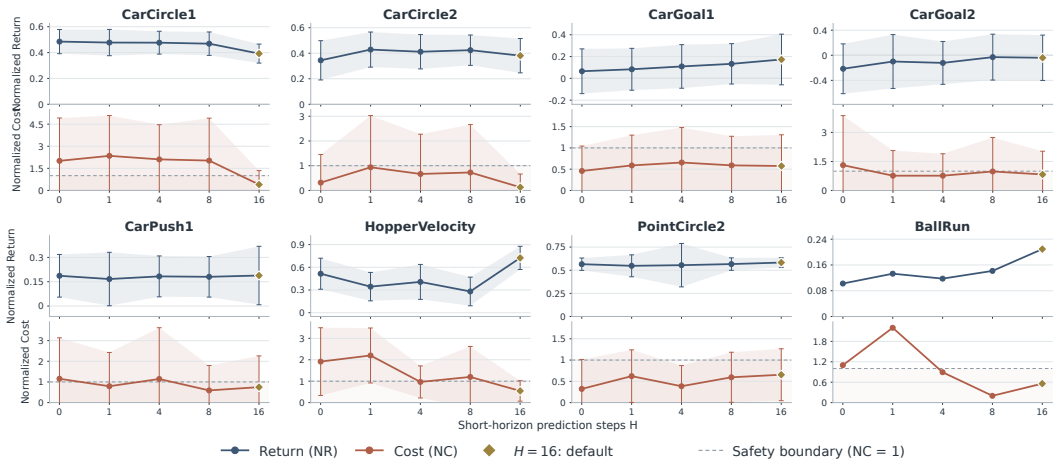

Figure 8: Sensitivity to the short-horizon scoring length H on eight illustrative tasks selected after a preliminary sweep. The upper and lower panels show normalized return (NR) and cost (NC). Points show means, and error bars and shading show episode-level standard deviations over 60 episodes. $H = 0$ omits short-horizon cost scoring; $H = 16$ reuses the main-table setting. The dashed line marks $\mathrm{NC} = 1$.

A.6 Limitations and Future Work

FIBA-Flow currently uses a fixed window for short-horizon cost prediction. Adaptive or multi-horizon prediction could offer greater flexibility for tasks with different cost delays. Our evaluation focuses on simulated benchmarks with a single cumulative-cost constraint. Future work could extend the method to multiple cost constraints and evaluate it in physical systems.