Can Agents Learn Safe Behavior From Non-Preferred Demonstrations?

Anonymous Authors
Paper under double-blind review

***Abstract*—Safe Reinforcement Learning(RL) applications, such as autonomous vehicles and robotic manipulation, require policies that avoid constraint violations while achieving task objectives. A key challenge is data scarcity: while large volumes of unlabeled operational data and identifiable failure cases are readily accessible, curated demonstrations that are both high-quality and verifiably safe are prohibitively scarce to collect. This data imbalance makes it essential to extract safe behaviors from heterogeneous datasets rather than relying exclusively on expert data. Existing methods, such as preference-based methods, suffer from cascading multi-stage errors, while safe imitation learning methods like SafeDICE require computationally expensive procedures to separate safe from unsafe behaviors. We propose Negative-Observation Preference Extraction (NOPE), a single-phase algorithm that integrates implicit preference learning with continuous-flow matching policy. NOPE leverages the Inverse Bellman Operator to extract implicit reward signals from preferences and weights the conditional flow matching objective with the resulting cumulative action-value estimates(Q values), incorporating safety constraints directly into the vector field without explicit reward models or the computational overhead of gradient-based guidance during inference. Experiments on navigation and velocity constraint tasks from the DSRL benchmark show that NOPE satisfies the safety constraints while achieving high returns. The dataset ablations confirm robustness to limited negative data and dataset heterogeneity. NOPE achieves high reward trajectories while being 1.64 times safer than the baselines on average.**

***Index Terms*—Inverse RL, Flow Matching, Preference Learning, Imitation Learning, Offline RL**

I. Introduction

Deploying reinforcement learning in real-world applications requires ensuring safety and avoiding unsafe behaviors whilst learning policies. In practice, obtaining high-quality expert demonstrations is expensive and time-consuming, yet large volumes of mixed-quality data are readily available: autonomous driving systems can access millions of hours of unlabeled footage and easily identify constraint-violating trajectories from incident reports, while verified safe demonstrations remain scarce. Learning safe policies directly from non-preferred, constraint-violating demonstrations paired with unlabeled mixed-quality data, without access to verified safe expert trajectories, therefore, remains a fundamental challenge. This data heterogeneity presents a structural barrier to standard imitation learning methods such as behavior cloning [1], which treat all observed behavior equally and reproduces undesirable actions when trained on mixed data.

Existing solutions such as preference-based methods [2], [3] address this by learning reward models from binary comparisons between trajectory segments, then optimizing a policy, which, however, leads to cascading multi-stage errors. Alternatively, safe offline RL approaches like SafeDICE [4] require computationally expensive density ratio estimation. Recent work on Inverse Preference Learning (IPL) [5] eliminates explicit reward modeling by learning value functions directly from preferences using the inverse Bellman operator, which establishes a bijection between preferences and value functions.

Concurrently, generative models have been adopted for imitation learning, with diffusion policies capturing multimodal behavior [6], [7]. However, iterative denoising introduces inference overhead, limiting real-time applicability [8]. Flow matching [9], [10] offers a more tractable alternative by regressing vector fields that generates a conditional probability path from noise to data, yielding a simpler training objective and faster inference compared to diffusion models [11]. Therefore, Flow matching is well-suited for policy learning enabling efficient, deterministic sampling in multimodal action spaces. Despite these advances, a key gap remains at the intersection of safety, preference learning, and generative control: enforcing safety constraints within Flow Matching without learning explicit reward models. Current guidance methods [12], [13] rely on auxiliary functions that reintroduce the alignment error and computational overhead of the two-phase paradigm.

In this work, we propose a single-phase algorithm that combines implicit reward learning through the inverse Bellman operator with energy-guided flow matching for policy generation. Our framework enables learning directly from mixed demonstrations, successfully distinguishing preferred trajectories from non-preferred trajectories without explicit reward engineering, density-ratio estimation, or separate training phases. Across six continuous control tasks spanning 2D navigation and high-dimensional MuJoCo locomotion, our method achieves high task returns while being $1.64\times$ safer than the baselines on average.

II. Related Works

A. Imitation Learning from Non-Preferred Demonstrations.

SafeDICE [4] learns safe behavior given a small negative set and a larger, unlabeled set. SafeDICE constructs a target policy as a mixture of an undesirable policy and a uniform random policy, it then adds a weight to the undesirable component and applies DICE optimization similar to DemoDICE [14].

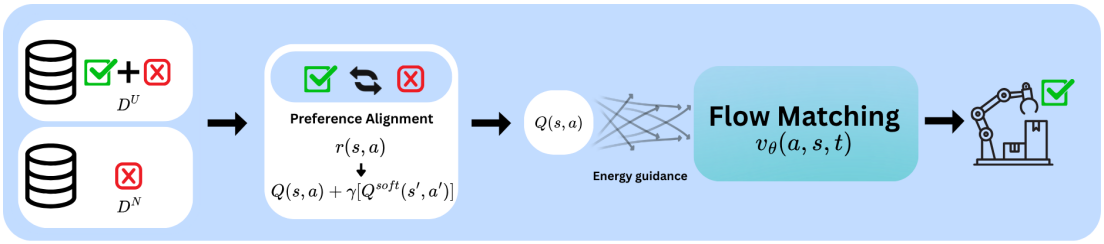

Fig. 1. Overview of the NOPE algorithm: Q-values learned from non-preferred demonstrations capture safety preferences, guiding a flow matching policy without explicit reward modeling or separate training phases.

B. Preference-Based Imitation Learning.

T-REX [15] and D-REX [16] are online algorithms that fit a reward model to ranked demonstrations. PEBBLE [3] and PrefPPO [17], also online learning, extend preference learning to human feedback through the Bradley–Terry model [18]. Offline counterparts such as OPRL [19] and methods from [20] replace the online RL step with offline policy optimisation, yet still require extensive pairwise annotations. SPRINQL [21] stratifies trajectories rather than the extensive pairwise comparisons. Crucially, all of these methods aim to *imitate* preferred trajectories, and rely on extra compute for training a reward model on the provided preferences.

Our algorithm simultaneously recovers an implicit reward signal and a deployable policy within a single Q-learning objective. Further, it does not make the assumption made in SafeDICE that the union set is largely composed of low-cost trajectories.

III. Preliminaries

We consider a Markov Decision Process (MDP) defined by the tuple $(\mathcal{S}, \mathcal{A}, p, r, \gamma)$ where $\mathcal{S}$ is the state space, $\mathcal{A}$ the action space, $p(s'|s,a)$ the transition dynamics, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ the reward function, and $\gamma \in [0,1)$ the discount factor. A policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ maps states to a probability distribution over actions. The Q-function $Q^\pi(s,a) = \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r(s_t,a_t)|s_0 = s, a_0 = a]$ denotes the expected return from taking action a in state s under policy π. The stationary distribution of a policy π is defined as: $d^\pi(s,a) = (1-\gamma)\sum_{t=0}^{\infty} \gamma^t \Pr(s_t = s, a_t = a \mid s_0 \sim p_0, \pi)$ where p_0 is the initial state distribution.

Our work addresses the offline setting where policies are learned from pre-collected trajectories lacking explicit reward or cost information, i.e., $\tau = (s_0, a_0, s_1, a_1, \ldots, s_{T-1}, a_{T-1}, s_T)$. We distinguish between preferred and non-preferred trajectories as follows:

Preferred Trajectory. A *preferred trajectory* achieves high returns while satisfying all safety constraints.

Non-preferred Trajectory. A *non-preferred trajectory* is defined as a trajectory sampled from the union of low returns or high cost trajectories.

We assume access to two datasets: a set of non-preferred trajectories $\mathcal{D}_N$ and a larger unlabeled union dataset $\mathcal{D}_U$ containing both preferred and non-preferred trajectories.

A. Inverse Bellman Operator

For a fixed policy π, the inverse Bellman operator $\mathcal{T}^\pi$ ([22]) establishes a bijection between the Q-function and rewards: $\mathcal{T}^\pi Q(s,a) = Q(s,a) - \gamma\, \mathbb{E}_{s' \sim p(\cdot|s,a)}[V^\pi(s')]$

This mapping is invertible since $(I - \gamma P^\pi)$ is non-singular for $\gamma < 1$, where P^π is the transition matrix under policy π. The inverse Bellman operator allows us to implicitly define rewards without explicitly learning a reward function.

B. Preference Learning

The Bradley-Terry model [18] provides a probabilistic framework for expressing preferences between trajectory segments. Given preference data $\mathcal{D}_p = \{(\tau^{(1)}, \tau^{(2)}, y)\}$ where $\tau = \{(s_t, a_t)\}_{t=0}^k$ are trajectory segments and $y \in \{0,1\}$ indicates preference, the probability that $\tau^{(1)}$ is preferred over $\tau^{(2)}$ is:

$$P[\tau^{(1)} \succ \tau^{(2)}] = \frac{\exp(R(\tau^{(1)}))}{\exp(R(\tau^{(1)})) + \exp(R(\tau^{(2)}))} \tag{1}$$

where $R(\tau) = \sum_{t=0}^{H-1} r(s_t, a_t)$ is the cumulative reward of a trajectory segment. Inverse Preference Learning (IPL) [5] substitutes the implicit reward $r = \mathcal{T}^\pi Q$ from the inverse Bellman operator into this model, enabling Q-functions to be optimized directly from preferences without an explicit reward function.

C. Flow Matching

Flow matching [9] provides a framework for learning probability paths through continuous normalizing flows (CNFs). FM models a time-dependent vector field $v_t(x)$ that pushes a simple source distribution p_0 (e.g., standard Gaussian) to a complex target data distribution p_1 (e.g., the policy distribution $\pi(a|s)$) via an Ordinary Differential Equation: $\frac{dx_t}{dt} = v_t(x_t), \quad x_0 \sim p_0(x)$.

To make training tractable, we utilize *Conditional Flow Matching* (CFM). Instead of matching the intractable marginal vector field, CFM regresses onto a conditional vector field $u_t(x|x_1)$ defined per data point $x_1 \sim p_1$. For a policy $\pi_\theta(a|s)$, the CFM objective is:

$$\mathcal{L}_{CFM}(\theta) = \mathbb{E}_{t\sim\mathcal{U},(s,a)\sim\mathcal{D},p_0(a_0)}\left[\|v_\theta(a_t,s,t) - u_t(a_t|a_1)\|^2\right] \tag{2}$$

where $a_t = (1-(1-\sigma_{min})t)a_0 + ta_1$ represents the optimal transport interpolation between noise and data. This provides a stable, regression-based objective for generative policy learning.

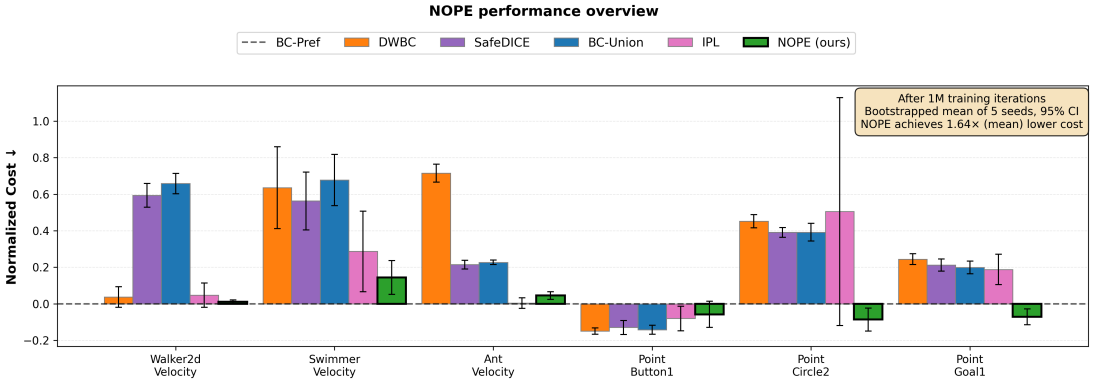

Fig. 2. NOPE consistently achieves the lowest cost across all tasks and is the only method to match or fall below the BC-Pref reference on navigation environments, achieving a $1.64\times$ reduction (mean) in normalized cost over the best-performing baseline per task. Notably, all methods achieve comparable normalized returns across tasks, confirming that NOPE's cost reduction is not obtained at the expense of task performance.

D. Energy Weighted Flow Matching for Offline RL

Energy-weighted flow matching [13] provides a framework for learning optimal policies by reweighting the conditional flow matching objective with Q-values. This approach enables direct learning of high-value behaviors without requiring test-time energy evaluation or classifier guidance:

$$\mathcal{L}_{EWFM}(\theta) = \mathbb{E}_{t,s,a_0,a_t}\left[\frac{e^{\beta Q_\psi(s,a_0)}}{\mathbb{E}_{\tilde{a}_0\sim\mu}\left[e^{\beta Q_\psi(s,\tilde{a}_0)}\right]}\|v_\theta - u_t^0\|^2\right] \quad (3)$$

where actions with higher Q-values receive proportionally higher weight during training through the importance weighting factor.

IV. Proposed Methodology

NOPE is a unified framework for learning safe policies from mixed-quality demonstrations without explicit reward engineering. Given non-preferred trajectories $\mathcal{D}_N$ and unlabeled trajectories $\mathcal{D}_U$, our method jointly trains preference-aligned Q-functions and energy-guided flow matching policies in a single loop.

Our approach combines three components: (1) learning Q-functions from preferences via the inverse Bellman operator (Section IV-A), (2) using Q-values to weight flow matching with entropy regularization for a more expressive policy (Section IV-B), and (3) combining these two components into a single joint training phase (Section IV-C). Algorithm 1 presents the complete training procedure.

A. Preference-Based Q-Learning

Preference-based learning has primarily been studied in settings where an annotator provides explicit pairwise comparisons between trajectory segments [2], [3], [5]. A key gap exists, however, in applying preference learning to the offline setting where no explicit labels are available and the dataset is a heterogeneous mixture of preferred and non-preferred trajectories. We address this by adapting the inverse preference learning framework to operate directly over non-preferred demonstrations and an unlabeled union dataset,

Sample segment pairs (τ^+, τ^-) from union/neg data
Sample flow segments $(s_{1:H}, a_{1:H})$ from union data
if step $>$ warmup **then**
  Compute preference loss $\mathcal{L}_Q$ (6) with soft Q-targets {Preference Learning}
  Update Q
  Compute segment energies $\mathcal{R}_i = \sum_h \gamma^h Q(s_{i,h}, a_{i,h})$
  $w_i \leftarrow \text{softmax}(\alpha_E \cdot \mathcal{R}_i)$
else
  $w_i \leftarrow 1/B$ (Uniform: pure behavior cloning)
end if
Compute weighted flow matching loss $\mathcal{L}_{\text{FM}}$ (9) {Flow Matching}
Update flow model v_θ

Fig. 3. NOPE Joint Training. During warmup (first 10^4 of 10^6 training iterations), uniform weights $w_i = 1/B$ enable pure behavior cloning, initializing the policy before Q-guidance. After warmup, segment energies $\mathcal{R}_i$ weight the flow loss toward high-value trajectories.

without requiring pairwise human annotations or an explicit reward model.

The Q-function learned by maximum entropy reinforcement learning algorithms implicitly encodes information about the reward function $r(s,a)$. Consequently, it is unnecessary to learn both explicitly. While the Q-function depends on both r and $\pi(a|s)$ under a given policy, this dependence can be disentangled through the soft Bellman equation. In the general stochastic case, the entropy term is retained and the operator takes the form:

$$\begin{aligned} r(s,a) &= (\mathcal{T}^\pi Q)(s,a) \\ &= Q(s,a) - \gamma\, \mathbb{E}_{\substack{s'\sim P(\cdot|s,a) \\ a'\sim\pi(\cdot|s')}}\Big[Q(s',a') - \alpha \log \pi(a'|s')\Big], \end{aligned} \quad (4)$$

where $a' = \pi(s')$ is the next action under the deterministic policy at successor state s', where $P(s'|s,a)$ denotes the

environment transition dynamics and $\alpha > 0$ controls the weight of the entropy bonus.

Constructing Preferences from Non-Preferred Demonstrations. The central contribution of this section is adapting preference-based Q-learning to operate without exhaustive pairwise annotations. Rather than requiring curated human comparisons, we construct a preference dataset $\mathcal{D}_p$ by treating all comparisons between unlabeled and non-preferred segments as preference pairs: for each unlabeled segment $\tau^U \in \mathcal{D}_U$ and non-preferred segment $\tau^N \in \mathcal{D}_N$, we create a preference pair $(\tau^U, \tau^N, y = 1)$ where $y = 1$ indicates that $\tau^U \succ \tau^N$ (unlabeled is chosen over non-preferred). This construction makes no assumption that the union set is predominantly composed of low-cost trajectories, which is a key limitation of prior methods such as SafeDICE [4].

Crucially, we substitute the implicit reward $r(s, a) = (\mathcal{T}^\pi Q)(s, a)$ from (4) into the Bradley-Terry model (1), allowing us to express preferences directly in terms of the Q-function. This yields the preference loss for learning Q:

$$\mathcal{L}_p(Q) = -\mathbb{E}_{(\tau^{(1)},\tau^{(2)},y)\sim\mathcal{D}_p}\Big[y \log P_Q[\tau^{(1)} \succ \tau^{(2)}] + (1-y)\log(1 - P_Q[\tau^{(1)} \succ \tau^{(2)}])\Big] \tag{5}$$

where $y \in \{0, 1\}$ indicates the ground-truth preference label (in our case, $y = 1$ when $\tau^{(1)} \in \mathcal{D}_U$ and $y = 0$ when $\tau^{(2)} \in \mathcal{D}_N$), and P_Q which comes from (1) denotes that preferences are computed using the Q-induced implicit reward. To ensure well-behaved implicit rewards and guarantee convergence, we follow IPL [5] in adding L_2 regularization on the implicit reward:

$$\mathcal{L}_Q = \mathcal{L}_p(Q) + \lambda \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[(\mathcal{T}^\pi Q(s, a))^2\right], \tag{6}$$

where $\lambda > 0$ controls the regularization strength [22]. The first term of (6) encourages the Q-function to match observed preferences, while the second term ensures the implied reward function is smooth, centered, and unique.

B. Energy-Weighted Flow Matching with Entropy Regularization

Having established how to learn a preference-aligned Q-function, we now leverage it to guide a flow matching policy towards high-value regions of the action space. As established in the Preliminaries, Energy-Weighted Flow Matching(EWFM) [13] provides a principled framework for reweighting the conditional flow matching objective by an energy function, steering the generative model towards a target distribution without requiring test-time gradient corrections. We adopt this energy weighting strategy at the segment level, using the cumulative Q-values from Section IV-A as the energy signal.

From Q-Values to Segment Weights. For each trajectory segment $\sigma_i = \{(s_{i,h}, a_{i,h})\}_{h=0}^{H-1}$ in the dataset, we compute the discounted cumulative Q-value over a horizon H (default = 10):

$$\mathcal{R}_i = \sum_{h=0}^{H-1} \gamma^h Q(s_{i,h}, a_{i,h}). \tag{7}$$

This segment-level estimate aggregates action quality over time, providing a holistic measure of trajectory desirability that goes beyond single-step Q-values. To convert segment returns into importance weights, we apply temperature-scaled softmax normalization:

$$w_i = \text{softmax}(\beta \cdot \mathcal{R}_i) = \frac{\exp(\beta \cdot \mathcal{R}_i)}{\sum_j \exp(\beta \cdot \mathcal{R}_j)}, \tag{8}$$

where $\beta > 0$ is the energy guidance temperature. The Q-weighted flow matching objective then combines conditional flow matching with Q-value-based importance weighting:

$$\mathcal{L}_{FM} = \sum_i w_i \frac{1}{|\mathcal{V}_i|} \sum_{h\in\mathcal{V}_i} \|v_\theta(a_{t,h}, s_{i,h}, t_h) - u_{t,h}(a_{t,h}|a_{i,h})\|^2, \tag{9}$$

where $\mathcal{V}_i$ is the set of valid timesteps in segment i, $v_\theta(a_t, s, t)$ is the learned vector field, and $a_t = (1-(1-\sigma_{\min})t)a_0 + ta_1$ is the optimal transport interpolation. Compared to standard EWFM which reweights individual transitions, our segment-based formulation encourages learning temporally coherent action sequences, which is critical for sequential decision-making.

Entropy Regularization. To couple the flow matching policy and the Q-function into a closed feedback loop (Fig. 1), we incorporate policy entropy directly into the Q-function update, allowing preference information to flow back into policy training and policy improvements to refine Q-estimates—all within a single phase. However, computing $\log \pi_\theta(a|s)$ for a flow matching policy requires integrating the vector field divergence along the ODE trajectory [23]:

$$\ln p_1(\psi_1(x)) = \ln p_0(\psi_0(x)) - \int_0^1 \nabla \cdot v(t, \psi_t(x))\, dt. \tag{10}$$

This incurs discretization error and Monte-Carlo variance [24], [25], which we address with the Skilling-Hutchinson trace estimator [26]:

$$\nabla \cdot \tilde{\mathbf{f}}_\theta(\mathbf{x}, t) = \mathbb{E}_{p(\boldsymbol{\epsilon})}\left[\boldsymbol{\epsilon}^\top \nabla \tilde{\mathbf{f}}_\theta(\mathbf{x}, t)\boldsymbol{\epsilon}\right], \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{11}$$

computed via reverse-mode autodiff at the cost of a single forward pass. With tractable $\log \pi_\theta(a|s)$, we define the soft Q-target:

$$Q^{\text{soft}}(s', a') = \hat{Q}(s', a') - \alpha \log \pi_\theta(a'|s'), \tag{12}$$

where $a' \sim \pi_\theta(\cdot|s')$. As the policy improves through energy-weighted training, its log-probability estimates sharpen the soft Q-targets, which yield more preference-aligned Q-values and energy weights for the next update. This soft Q-function is substituted into the inverse Bellman operator from Section IV-A:

$$r(s, a) = Q(s, a) - \gamma\, \mathbb{E}_{\substack{s'\sim P(\cdot|s,a)\\ a'\sim\pi(\cdot|s')}}\Big[\hat{Q}(s', a') - \alpha \log \pi_\theta(a'|s')\Big] \tag{13}$$

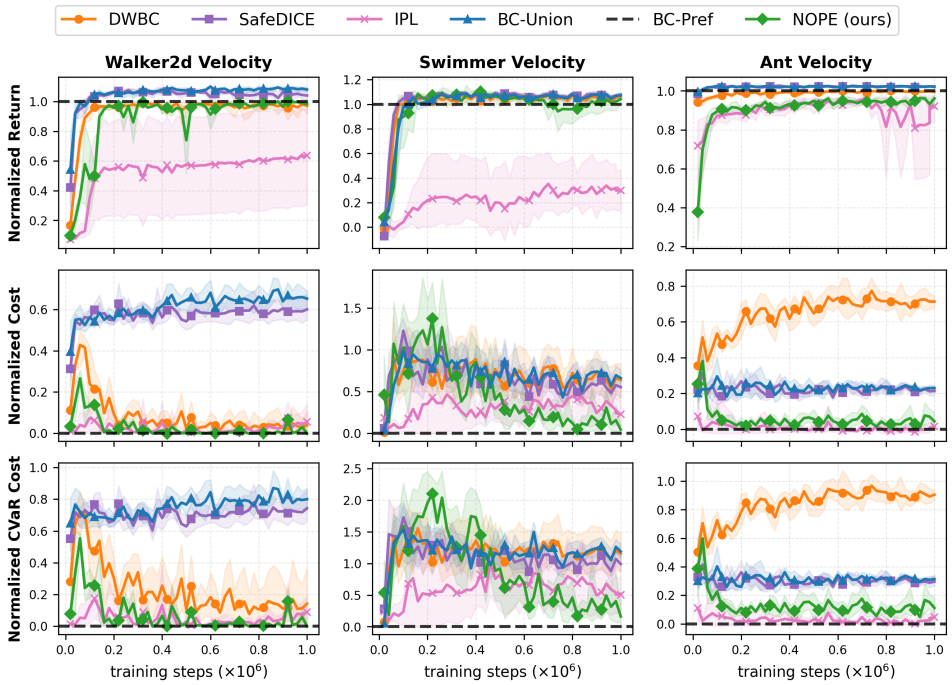

Fig. 4. Comparative performance on continuous locomotion tasks (`Walker2d-Velocity`, `Swimmer-Velocity`, `Ant-Velocity`). NOPE performs significantly better than baselines. Comparing NOPE with IPL, IPL also displays low cost behavior, however, policy learning is unreliable, judging from the high variance in eval results, and IPL policy failing to deliver high returns.

C. Joint Single-Phase Training

A central contribution of NOPE is the elimination of the two-phase paradigm that afflicts existing approaches. Prior methods first learn an explicit reward model from preference labels, then separately optimize a policy against it, cascading errors from the first stage into the second [2], [4]. NOPE discards explicit reward modeling entirely by coupling two complementary mechanisms within a single unified training loop:

1) **Preference-Aligned Q-Learning:** The Q-function is trained directly on trajectory preference structure—non-preferred and unlabeled trajectories—learning which behaviors the data implicitly favors without fitting any intermediate reward signal.
2) **Energy-Weighted Flow Matching:** Segment-level Q-values serve as energy weights that bias the flow matching objective toward high-preference regions of trajectory space, steering the learned vector field away from non-preferred behaviors.

These two components reinforce each other through a closed feedback loop: as the Q-function becomes better aligned with preference structure, it produces sharper energy weights that focus flow matching on preferred trajectories; as the policy improves, its entropy feeds back into the soft Q-target, keeping both components tightly coupled throughout training. The result is a single-phase algorithm that recovers preference-aligned behavior and a deployable policy simultaneously - without explicit reward modeling, density-ratio estimation, or separate training phases.

V. Experiments

We focus our experiment efforts to answer the following questions: **Does** NOPE perform better compared to prior baselines? **How** does NOPE react to different sizes of data classes, namely the unlabeled set $\mathcal{D}_U$ and non-preferred set $\mathcal{D}_N$?

A. Experimental Setup

Environments and Datasets. We evaluate our approach using the Datasets for Safe Reinforcement Learning (DSRL) benchmark [27]. We selected 6 DSRL tasks to comprehensively evaluate NOPE: three navigation tasks featuring spatial safety constraints (`Point-Goal1`, `Point-Button`, `Point-Circle2`) and three locomotion tasks featuring kinematic velocity constraints (`Ant-Velocity`, `Swimmer-Velocity`, `Walker2d-Velocity`).

We construct dataset $\mathcal{D}_N$ by sampling a limited number of trajectories from the non preferred set. The set $\mathcal{D}_U$ is made by sampling from sets containing non-preferred trajectories as well as preferred trajectories. For evaluation purposes, we also compose a dataset $\mathcal{D}_P$, which contains only the preferred trajectories that compose the set $\mathcal{D}_U$. It is ensured that the set $\mathcal{D}_U$ is always larger than the set $\mathcal{D}_P$. We mask the cost and reward labels before sampling and constructing datasets.

Baselines. We compare NOPE against four baselines: 1.) Behavior Cloning on the union set ($\mathcal{D}_U$) **BC-Union**: serves as

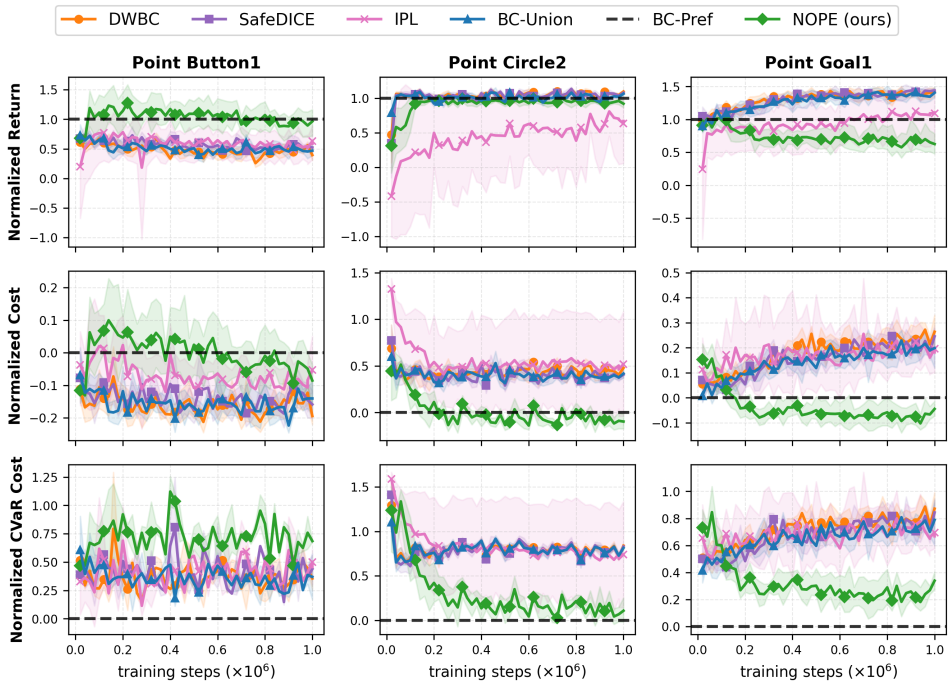

Fig. 5. Comparative performance on spatial navigation tasks (`Point-Button1`, `Point-Circle2`, `Point-Goal1`). NOPE is effective for all navigation tasks. Notice that NOPE has higher cost for PointButton1 task, however, it also achieves much higher reward compared to the baselines, its performance lingering close to the BC-Pref baseline, thereby performing better than baselines

a baseline to evaluate whether the policy is learning from the non-preferred set. 2.) **DWBC** ([28]): this is weighted behavior cloning; the weights are learn via a discriminator trained via *non-preferred learning*. 3.) **SafeDice**([4]): this is a state of the art framework for such problems. SafeDice minimizes occupancy overlap with D_N via density-ratio estimation. 4.) **IPL** ([5]): IPL implementation with IQL actor in this setting to evaluate the benefits of a flow matching policy in this setting.

Evaluation. We report the **normalized return**, **normalized cost** and **normalized** $CVaR_{20}$ **cost**. 1.) **Normalized Return** scales the mean episodic returns with 0 as the minimum possible episodic return and 1 as the episodic return for a BC policy strictly trained on the preferred trajectory set $\mathcal{D}_P$ (BC-Pref). 2.) **Normalized Cost** scales the mean episodic costs with 0 as BC-Pref costs and 1 as the maximum possible episodic costs. 3.) $CVaR_{20}$ **Cost** scales the policy's mean episodic cost of the worst 20% runs. All plots show averages over 50 evaluation episodes. To assess statistical significance, we generate 1000 bootstrap samples from the data, using results from 5 random seeds, plotting 95% confidence intervals.

B. Does NOPE perform better compared to prior baselines?

As observed in Fig. 4 and Fig. 5, NOPE consistently generates high-reward, low-cost trajectories across both velocity and navigation tasks. For **PointButton1**, NOPE achieves a higher reward than all other baselines, at the cost of moderately higher constraint violations, tracking closely with the BC-Pref line. The dataset composition explains this trade-off: the majority of trajectories in this environment are low-cost but also low-reward, causing baselines to collapse toward the BC-Union line rather than learning preference-aligned behavior. NOPE, successfully extracts the preference signal and targets the high-reward region, consistent with the BC-Pref upper bound. For the remaining navigation tasks, NOPE significantly outperforms them, achieving high returns that effectively follow constraints.

C. How does NOPE react to different sizes of data classes?

Further experiments and tests reveal NOPE maintains performance when D_N size is reduced.

VI. Conclusion

We presented Negative-Observation Preference Extraction (NOPE), a single-phase framework for learning safe policies directly from constraint-violating demonstrations and unlabeled data. By combining implicit preference learning via the inverse Bellman operator with energy-weighted flow matching, NOPE eliminates the multi-stage error propagation and density-ratio estimation overhead. Preference information is directly incorporated into the Q-function, whose segment-level values bias the flow-matching objective toward preferred behaviors within a closed feedback loop that requires no explicit reward modeling or separate training phases. Empirical evaluation across six continuous control tasks demonstrates that NOPE achieves strong returns while substantially reducing constraint violations. Preliminary experiments suggest that NOPE is robust to limited negative data and unlabeled operational data, opening pathways for safer reinforcement learning in safety-critical applications.

REFERENCES

[1] D. A. Pomerleau, "Efficient training of artificial neural networks for autonomous navigation," *Neural Computation*, vol. 3, no. 1, pp. 88–97, 1991.

[2] P. Christiano, J. Leike, T. B. Brown, M. Martic, S. Legg, and D. Amodei. (2023) Deep reinforcement learning from human preferences. unpublished. [Online]. Available: https://arxiv.org/abs/1706.03741

[3] K. Lee, L. Smith, and P. Abbeel. (2021) PEBBLE: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. unpublished. [Online]. Available: https://arxiv.org/abs/2106.05091

[4] Y. Jang, G.-H. Kim, J. Lee, S. Sohn, B. Kim, H. Lee, and M. Lee, "SafeDICE: Offline safe imitation learning with non-preferred demonstrations," in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, vol. 36, 2023, pp. 74 921–74 951.

[5] J. Hejna and D. Sadigh. (2023) Inverse preference learning: Preference-based RL without a reward function. unpublished. [Online]. Available: https://arxiv.org/abs/2305.15363

[6] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song. (2024) Diffusion policy: Visuomotor policy learning via action diffusion. unpublished. [Online]. Available: https://arxiv.org/abs/2303.04137

[7] T. Pearce, T. Rashid, A. Kanervisto, D. Bignell, M. Sun, R. Georgescu, S. V. Macua, S. L. Tan, I. Momennejad, K. Hofmann, and S. Berns, "Imitating human behaviour with diffusion models," *Proc. International Conference on Learning Representations (ICLR)*, 2023.

[8] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Elucidating the design space of diffusion-based generative models," in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, 2022.

[9] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le. (2023) Flow matching for generative modeling. unpublished. [Online]. Available: https://arxiv.org/abs/2210.02747

[10] X. Liu, C. Gong, and Q. Liu, "Flow straight and fast: Learning to generate and transfer data with rectified flow," *Proc. International Conference on Learning Representations (ICLR)*, 2023.

[11] M. S. Albergo, N. M. Boffi, and E. Vanden-Eijnden. (2023) Stochastic interpolants: A unifying framework for flows and diffusions. unpublished. [Online]. Available: https://arxiv.org/abs/2303.08797

[12] W. Fan, A. Y. Zheng, R. A. Yeh, and Z. Liu. (2025) CFG-Zero*: Improved classifier-free guidance for flow matching models. unpublished. [Online]. Available: https://arxiv.org/abs/2503.18886

[13] S. Zhang, W. Zhang, and Q. Gu. (2025) Energy-weighted flow matching for offline reinforcement learning. unpublished. [Online]. Available: https://arxiv.org/abs/2503.04975

[14] G. hyeong Kim, S. Seo, J. Lee, W. Jeon, H. Hwang, H. Yang, and K.-E. Kim, "DemoDICE: Offline imitation learning with supplementary imperfect demonstrations," in *Proc. International Conference on Learning Representations (ICLR)*, 2022.

[15] D. Brown, W. Goo, P. Nagarajan, and S. Niekum. (2019) Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. unpublished. [Online]. Available: https://arxiv.org/abs/1904.06387

[16] D. S. Brown, W. Goo, and S. Niekum. (2020) Better-than-demonstrator imitation learning via automatically-ranked demonstrations. unpublished. [Online]. Available: https://arxiv.org/abs/1907.03976

[17] K. Metcalf, M. Sarabia, and B.-J. Theobald. (2022) Rewards encoding environment dynamics improves preference-based reinforcement learning. unpublished. [Online]. Available: https://arxiv.org/abs/2211.06527

[18] R. A. Bradley and M. E. Terry, "Rank analysis of incomplete block designs: I. The method of paired comparisons," *Biometrika*, vol. 39, p. 324, 1952. [Online]. Available: https://api.semanticscholar.org/CorpusID:125209808

[19] D. Shin, A. D. Dragan, and D. S. Brown. (2023) Benchmarks and algorithms for offline preference-based reward learning. unpublished. [Online]. Available: https://arxiv.org/abs/2301.01392

[20] C. Kim, J. Park, J. Shin, H. Lee, P. Abbeel, and K. Lee. (2023) Preference Transformer: Modeling human preferences using Transformers for RL. unpublished. [Online]. Available: https://arxiv.org/abs/2303.00957

[21] H. Hoang, T. Mai, and P. Varakantham. (2024) SPRINQL: Sub-optimal demonstrations driven offline imitation learning. unpublished. [Online]. Available: https://arxiv.org/abs/2402.13147

[22] D. Garg, S. Chakraborty, C. Cundy, J. Song, M. Geist, and S. Ermon. (2022) IQ-Learn: Inverse soft-Q learning for imitation. unpublished. [Online]. Available: https://arxiv.org/abs/2106.12142

[23] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural ordinary differential equations," in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, vol. 31, 2018.

[24] M. F. Hutchinson, "A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines," *Communications in Statistics - Simulation and Computation*, vol. 18, no. 3, pp. 1059–1076, 1989.

[25] T. Sauer, *Numerical Analysis*. Addison-Wesley, 2011.

[26] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," in *Proc. International Conference on Learning Representations (ICLR)*, 2021.

[27] Z. Liu, Z. Guo, H. Lin, Y. Yao, J. Zhu, Z. Cen, H. Hu, W. Yu, T. Zhang, J. Tan, and D. Zhao. (2023) Datasets and benchmarks for offline safe reinforcement learning. unpublished. [Online]. Available: https://arxiv.org/abs/2306.09303

[28] H. Xu, X. Zhan, H. Yin, and H. Qin. (2022) Discriminator-weighted offline imitation learning from suboptimal demonstrations. unpublished. [Online]. Available: https://arxiv.org/abs/2207.10050