Negative Energy as Reward: Optimizing Beyond Demonstrations in Offline Goal-Conditioned Control

Carlos Vélez García
Robotics & Automation
INESCOP
Elda, Alicante, Spain
cvelez@inescop.es

Miguel Cazorla
University Institute of Computing Research
University of Alicante
Alicante, Spain
miguel.cazorla@ua.es

Jorge Pomares
University Institute of Computing Research
University of Alicante
Alicante, Spain
jpomares@ua.es

***Abstract*—Imitation Learning (IL) enables training robot policies from demonstrations but is limited by distribution shift and by its inability to improve beyond demonstrator quality. Reinforcement learning (RL) addresses these limitations through objective optimization, yet integrating such optimization into reward-free offline IL remains challenging.**

We introduce NEaR (Negative Energy as Reward), a reconstruction-based trajectory energy model trained solely from demonstrations. Although learned via masked denoising, the model defines a scalar energy whose gradient enables constrained goal-directed refinement at inference. Planning is formulated as energy minimization, interpreting negative energy as a reward proxy without explicit reward supervision or temporal-difference learning.

On the `cube-single` task from OGBench, NEaR achieves 93% success on the structured `play` dataset and 98% on the higher-dispersion noisy dataset, closely matching goal-conditioned value-based methods. Diagnostic analyses indicate that increased dataset dispersion reshapes the learned energy landscape, improving refinement under distribution shift.

These results highlight objective optimization as a key mechanism through which RL can enhance IL-trained robot policies.

***Index Terms*—Imitation Learning, Offline Reinforcement Learning, Energy-Based Models, Goal-Conditioned Control, Trajectory Optimization, Reward Learning**

I. Introduction

Imitation learning (IL) [1] has become a dominant paradigm for training robotic policies from demonstrations [2]–[6]. While effective in many settings, IL often remains limited by distribution shift and by its inability to systematically improve beyond the quality of the demonstrations. Reinforcement learning (RL) [7], in contrast, optimizes behavior with respect to an objective and can in principle surpass demonstrator performance. Understanding how optimization mechanisms from RL can complement imitation learning—especially in offline and reward-free settings—remains an open challenge.

Recent offline goal-conditioned benchmarks reveal a striking phenomenon: suboptimal and noisier datasets can yield higher goal-reaching performance than cleaner and more structured demonstrations [8], [9]. In OGBench, value-based methods benefit substantially from higher-coverage noisy data, even when individual trajectories are suboptimal. These observations suggest that the presence of an optimizable objective—rather than strict action imitation—plays a central role in robust goal-conditioned performance.

In this work, we investigate how optimization—central to reinforcement learning—can complement imitation learning without requiring explicit reward supervision. We propose NEaR (Negative Energy as Reward), a reconstruction-based trajectory energy-based model trained solely on reward-free demonstration data. Although learned through imitation-style masked denoising reconstruction, the energy defines a scalar objective whose gradient enables trajectory refinement under boundary constraints. In this sense, NEaR augments imitation with an explicit optimization mechanism, enabling trajectory improvement beyond direct behavioral cloning. Figure 1 provides a conceptual illustration of how increased trajectory dispersion shapes the learned energy landscape and supports stable gradient-based refinement.

NEaR can be interpreted as a minimal form of reward learning: it derives an optimizable objective directly from demonstrations, without preference labels or external rewards. Importantly, NEaR does not propose a new RL algorithm or replace value learning. Rather, it isolates the role of objective optimization within a purely imitation-trained model, providing a controlled setting to examine how introducing an optimizable objective can enhance robustness and generalization in offline goal-conditioned control.

Empirically, we show that this reconstruction-based objective captures performance trends associated with value-based methods and offers geometric insight into why increased dataset dispersion improves goal-reaching behavior. Together, these results highlight objective optimization as a key mechanism through which reinforcement learning can complement imitation learning.

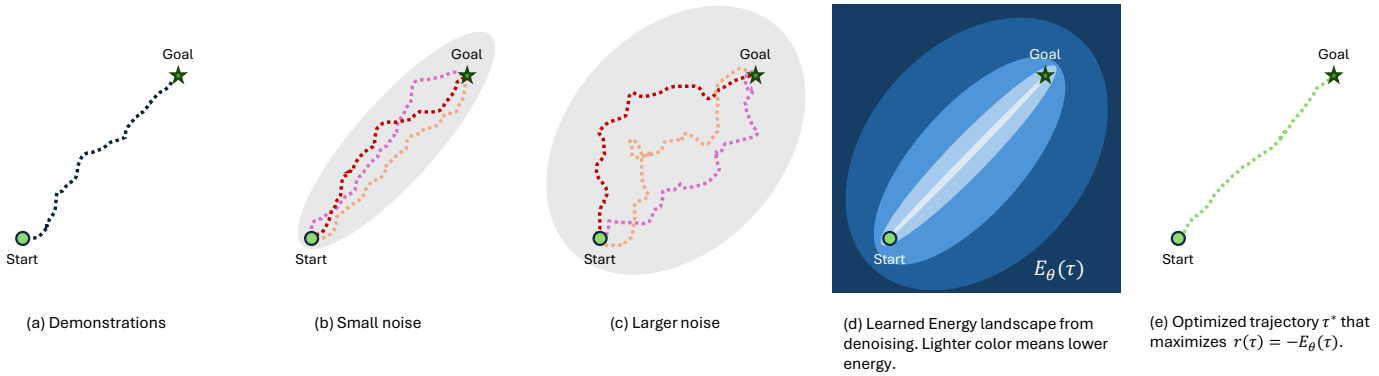
This perspective offers a minimal and interpretable bridge between imitation learning and reinforcement learning.

II. Related Work

A. Imitation Learning and Generative Policies

Imitation learning aims to reproduce expert behavior directly from demonstrations [1]. Behavioral cloning learns a mapping from observations to actions but is limited by

This work was funded by the Centre for the Development of Industrial Technology (CDTI), Grant No. CER 20251010. ROCCA Project.



(a) Demonstrations (b) Small noise (c) Larger noise (d) Learned Energy landscape from denoising. Lighter color means lower energy. (e) Optimized trajectory τ^* that maximizes $r(\tau) = -E_\theta(\tau)$.

Fig. 1. **NEaR high-level intuition**: given a demonstration from a fixed dataset of trajectories (a), we generate corrupted sub-trajectories by masking and injecting noise (b–c). The model learns an energy landscape over trajectories through iterative denoising, where lighter regions correspond to lower energy and higher data consistency (d). At inference, goal-directed planning is performed by initializing masked states with noise and iteratively following the negative energy gradient under boundary constraints (start and goal fixed). This refinement produces a trajectory $\tau^\star$ that minimizes the learned energy (equivalently maximizes $r(\tau) = -E_\theta(\tau)$), yielding a plausible goal-reaching plan (e).

distribution shift and by the quality of the dataset. Recent advances leverage expressive sequence models and generative architectures to scale imitation learning, including transformer-based policies [4], [5] and diffusion-based action models [3], [6]. While these approaches improve robustness and scalability, they remain fundamentally imitation-based: policies are trained to match demonstrated actions rather than to optimize an explicit objective at inference time.

B. Reward Learning and Beyond-Demonstrator Optimization

Reinforcement learning enables improvement beyond demonstrator performance by optimizing an explicit reward function. When ground-truth rewards are unavailable, reward learning methods seek to infer an objective from demonstrations. T-REX [10] learns a reward model from ranked trajectory comparisons, enabling policy optimization without access to true rewards. D-REX [11] extends this framework by injecting noise into demonstrations to improve reward identifiability and generalization. In both cases, improvement beyond the demonstrator arises from learning an objective that can be optimized independently of the original action distribution. More recently, methods such as Self-Supervised Reward Regression (SSRR) [12] model the relationship between policy degradation and injected noise to learn reward functions from suboptimal demonstrations.

NEaR differs from preference-based reward learning in that it does not rely on trajectory rankings or explicit reward inference. Instead, it learns a trajectory energy via reconstruction and interprets its negative as a reward proxy. This provides a minimal mechanism through which optimization can complement imitation without requiring preference labels or external supervision.

C. Offline Goal-Conditioned Reinforcement Learning

Value-based offline reinforcement learning methods such as CQL [13] and IQL [14] estimate objectives that enable policy improvement from static datasets. Goal-conditioned extensions of these approaches, such as GCIQL, GCIVL [8], have demonstrated strong performance on recent benchmarks like OGBench where behavioral cloning GCBC [2] fails, highlighting the importance of objective optimization in reward-free settings [9].

D. Energy-Based and Gradient-Based Planning

Energy-Based Models (EBMs) provide a flexible framework for modeling dependencies between variables by associating a scalar *energy* –a measure of compatibility– to each configuration [15]. Given a configuration, inference consists of finding a compatible solution by minimizing the learned energy. This formulation defines an explicit objective landscape over trajectories and enables direct gradient-based optimization in trajectory space.

In robotics and control, related ideas have been explored for trajectory optimization and structured planning. Diffusion-based planners [16] iteratively refine trajectories using learned denoising or score functions, typically combined with external reward guidance. While diffusion methods can also employ deterministic samplers [17], their inference procedure approximates sampling from a learned generative process rather than directly minimizing a scalar objective defined over trajectories.

In contrast, NEaR explicitly learns a scalar energy over trajectories from demonstrations. Inference is formulated as constrained energy minimization, where the negative energy serves as a reward proxy and its gradient defines the refinement direction. Unlike diffusion-based planners, which approximate sampling from a learned generative process, NEaR directly minimizes a learned scalar objective over trajectories. Optimization –not sampling– is the primary inference mechanism.

III. Method

We present NEaR (Negative Energy as Reward), a reconstruction-based trajectory-level energy learning framework for offline goal-conditioned control. NEaR learns an energy function over trajectories from unlabeled data and performs planning through gradient-based refinement under boundary constraints.

Unlike reward regression or preference-based approaches, NEaR does not assume access to reward labels, goal-conditioned supervision, or pairwise comparisons. Instead, it learns an energy over trajectories via masked denoising and uses the gradient of this energy to refine candidate trajectories at inference.

A. Trajectory Energy Learning

Let $\mathcal{D} = \{\tau^{(n)}\}_{n=1}^{N}$ denote a static dataset of trajectories, where each trajectory $\tau = (s_0, \ldots, s_T)$ consists of states $s_t \in \mathbb{R}^d$. During training, we sample fixed-length sub-trajectories

$$\mathbf{s} = (s_{t_0}, \ldots, s_{t_0+L}). \tag{1}$$

which we denote as the ground-truth trajectory $\mathbf{s}_{\text{true}}$ for reconstruction.

To enable partial reconstruction, we define a binary mask $M \in \{0,1\}^L$ indicating which states may be modified. A prefix of the trajectory and one randomly selected future state are fixed, while the remaining states are masked.

We parameterize an energy function

$$E_\theta(\mathbf{s}; M), \tag{2}$$

implemented as a non-causal decoder-only transformer.

B. Corruption and Gradient-Based Refinement

Masked states of $\mathbf{s}_{\text{true}}$ are corrupted with Gaussian noise:

$$\tilde{\mathbf{s}} = \sqrt{\alpha}\,\mathbf{s} + \sqrt{1-\alpha}\,\epsilon, \tag{3}$$

where $\epsilon \sim \mathcal{N}(0, I)$ and $\alpha \in (0,1)$ is a sampled corruption level from a logit-normal distribution. Unmasked states remain fixed.

Starting from $\tilde{\mathbf{s}}$, we perform T refinement steps by following the negative gradient of the energy with respect to the trajectory:

$$\mathbf{s}^{k+1} = \mathbf{s}^k - \eta \nabla_{\mathbf{s}} E_\theta(SG(\mathbf{s}^k); M), \tag{4}$$

where η is a fixed step size hyperparameter and SG denotes the *stop-gradient* operation that prevents backpropagation-through-time. Empirically, this stop-gradient operation stabilizes optimization and accelerates convergence. The gradient $\nabla_{\mathbf{s}} E_\theta$ defines a vector field over trajectory space, indicating directions that decrease energy.

After each refinement step, boundary constraints are enforced:

$$\mathbf{s}^{k+1} = (1-M) \odot \mathbf{s}_{\text{true}} + M \odot \mathbf{s}^{k+1}. \tag{5}$$

Thus, only masked states are updated while observed states remain fixed.

The model is trained to reconstruct masked states across refinement steps using a Smooth L1 loss applied only to masked elements:

$$\mathcal{L}_{\text{NEaR}} = \sum_{k=1}^{T} \left\| \mathbf{s}^k[M] - \mathbf{s}_{\text{true}}[M] \right\|_1. \tag{6}$$

Where $[M]$ indexes the masked states. This objective shapes the energy so that its gradient points toward the trajectory manifold represented in the dataset.

C. Inverse Dynamics

To convert refined trajectories into executable actions, we train an inverse dynamics model f_ϕ that predicts

$$a_t = f_\phi(s_t, s_{t+1}), \tag{7}$$

using supervised regression on dataset transitions.

The overall training objective is

$$\mathcal{L} = \mathcal{L}_{\text{NEaR}} + \mathcal{L}_{\text{action}}, \tag{8}$$

where $\mathcal{L}_{\text{action}}$ denotes the regression loss for inverse dynamics.

D. Goal-Directed Inference as Constrained Optimization

At inference time, NEaR performs goal-directed planning through gradient-based refinement under boundary constraints. Given the current state s_t and a desired goal state s_G, we construct candidate trajectories by fixing boundary conditions: states up to time t are clamped to the observed trajectory, and one future index t_g is clamped to the goal state. The remaining states are masked.

Masked states are initialized as pure Gaussian noise,

$$\tilde{\mathbf{s}} = \epsilon, \quad \epsilon \sim \mathcal{N}(0, I), \tag{9}$$

while boundary states remain fixed. This corresponds to the corruption model in Eq. 3 with $\alpha = 0$ at inference.

Planning is then formulated as constrained energy minimization,

$$\min_{\tau} E_\theta(\tau) \quad \text{subject to} \quad s_0 = s_t,\ s_{t_g} = s_G, \tag{10}$$

where τ denotes the candidate trajectory.

We approximately solve (10) via iterative gradient descent,

$$\mathbf{s}^{k+1} = \mathbf{s}^k - \eta \nabla_{\mathbf{s}} E_\theta(\mathbf{s}^k; M), \tag{11}$$

with boundary constraints re-imposed after each step. The gradient $\nabla_{\mathbf{s}} E_\theta$ defines a vector field over trajectory space, guiding refinement toward low-energy trajectories consistent with the learned data manifold.

Multiple candidate trajectories are refined in parallel with different t_g. After refinement, the trajectory with minimum energy is selected,

$$\tau^\star = \arg\min_{\tau} E_\theta(\tau), \tag{12}$$

and the next action is obtained via inverse dynamics,

$$a_t = f_\phi(s_t, s_{t+1}^\star). \tag{13}$$

NEaR replans in a receding-horizon manner every K steps.

E. Interpretation

Although trained via reconstruction, the learned energy defines a scalar objective over trajectories. We interpret

$$r(\tau) = -E_\theta(\tau) \tag{14}$$

as a reward proxy. Planning corresponds to maximizing this learned objective under boundary constraints. Importantly, the gradient of the energy function determines the refinement dynamics, thereby defining the geometry that guides goal-directed behavior.

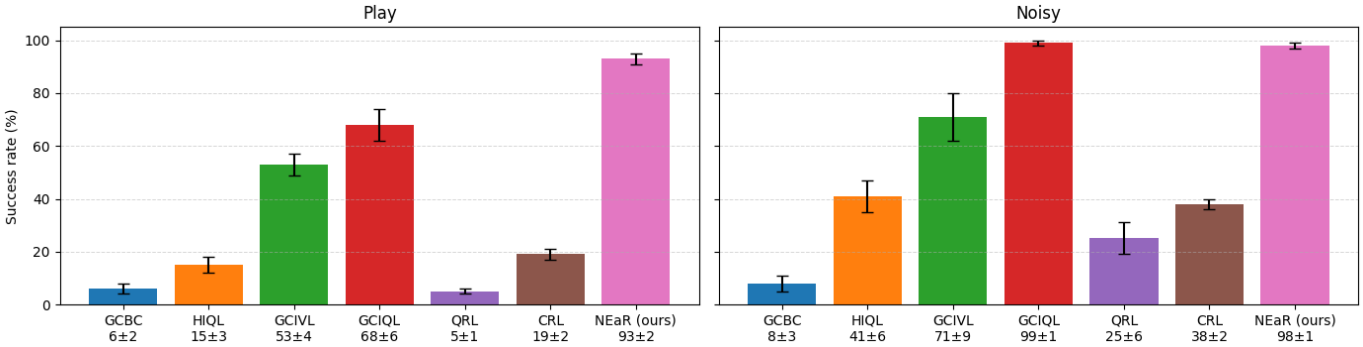


Fig. 2. **Success rates on `cube-single` under `play` and `noisy` datasets.** Comparison of NEaR with imitation-based (GCBC) and representative goal-conditioned value-based baselines. On `play`, NEaR achieves $93\% \pm 2$, substantially outperforming behavioral cloning and exceeding value-based methods. On `noisy`, NEaR reaches $98\% \pm 1$, closely matching GCIQL ($99\% \pm 1$). Results are averaged over evaluation seeds; error bars denote standard deviation.

IV. Experiments

A. Experimental Setup

We evaluate NEaR on the `cube-single` task from OGBench [8], a benchmark designed to study offline goal-conditioned reinforcement learning under reward-free and heterogeneous data conditions. We use the *state-based observation variant* of the environment, where policies observe low-dimensional proprioceptive and object state features rather than images.

The `cube-single` task requires controlling a robotic manipulator to move a cube to arbitrary target poses. Fig. 2 reports the success rates on the `play` and `noisy` dataset variants. These variants differ in data collection dynamics:

Play dataset (`play`). Collected using open-loop, non-Markovian expert policies with temporally correlated noise. These trajectories exhibit naturalistic, smooth behavior but relatively narrow state coverage.

Noisy dataset (`noisy`). Collected using closed-loop, Markovian expert policies with larger, uncorrelated Gaussian noise. These trajectories are less smooth and suboptimal, often failing grasps or dropping the cube between pick-and-place goals, resulting in significantly broader state dispersion.

Importantly, the `play` dataset contains behavior that appears more structured and coherent, whereas the `noisy` dataset sacrifices per-trajectory optimality in exchange for increased coverage of the state space. Prior benchmark results indicate that several goal-conditioned RL methods achieve substantially higher performance on the higher-coverage `noisy` dataset despite its lower trajectory quality [8], [9].

This contrast makes `cube-single` particularly well-suited for analyzing how dataset dispersion influences reward learning and optimization. In this work, we examine whether a trajectory energy learned purely from imitation data can exploit broader state coverage to enable improved goal-directed refinement beyond behavioral cloning.

We follow the standard OGBench evaluation protocol. Policies are evaluated on five predefined target configurations, and success rate is reported as the percentage of episodes that reach the goal within the fixed time horizon. All methods are trained using identical dataset splits for comparability.

B. Baseline Performance Across Play and Noisy Variants

We first examine the performance of established goal-conditioned imitation and reinforcement learning methods on the `cube-single` task.

The results reveal a clear divergence between imitation-based and value-based approaches. Pure behavioral cloning (GCBC) performs poorly on both datasets. On `play`, the narrow and structured trajectory distribution results in a brittle policy: small deviations from the training distribution cannot be corrected at test time, leading to cascading errors. On `noisy`, performance remains limited because behavioral cloning is inherently upper-bounded by demonstration quality and tends to reproduce suboptimal behavior.

In contrast, goal-conditioned Q-learning methods such as GCIQL exhibit a markedly different trend. While performance on `play` remains moderate, success increases dramatically on `noisy` data, reaching near-perfect performance. This suggests that value-based methods benefit from increased state coverage, which improves the learned objective landscape and enables optimization at inference time to better recover from distribution shift.

These observations highlight an important empirical phenomenon: suboptimal but higher-variance data can lead to superior goal-reaching performance when optimization is performed over a learned objective. From the perspective of imitation learning alone, this is counterintuitive.

NEaR is designed to bridge these paradigms. Like behavioral cloning, it is trained through reconstruction on offline data without reward labels. However, at inference it performs optimization over a learned energy, effectively behaving as a reward-based planner. In the following sections, we investigate whether this reconstruction-based reward proxy can reproduce the performance trends observed in value-learning methods and provide insight into the role of dataset dispersion.

C. NEaR Performance Across Play and Noisy Variants

Figure 2 reports success rates compared to representative imitation-based and value-based baselines.

On the `play` dataset, NEaR achieves $93\% \pm 2$, substantially outperforming behavioral cloning (GCBC: $6\% \pm 2$) and improving upon goal-conditioned RL methods. This indicates that introducing an optimizable objective into imitation-trained models can potentially mitigate brittleness arising from narrow trajectory distributions.

On the `noisy` dataset, NEaR achieves $98\% \pm 1$, closely matching the performance of goal-conditioned value learning (GCIQL: $99\% \pm 1$). Importantly, this performance is obtained without temporal-difference learning or explicit reward supervision. Instead, improvement arises from gradient-based refinement over the learned energy landscape.

Together, these results exhibit the same qualitative trend observed in value-based methods: broader state dispersion leads to improved goal-reaching performance when an optimizable objective is available. NEaR benefits from broader state coverage by enabling more effective gradient-based refinement at inference time.

These results indicate that introducing an optimizable objective into imitation-trained models can reproduce the performance trends observed in value-based methods under varying dataset dispersion.

V. Dataset Dispersion Shapes Energy Geometry and Refinement Dynamics

The experiments above establish two empirical observations:

(i) NEaR reproduces the performance trend of value-based methods across dataset variants (Fig. 2), and (ii) models trained on the higher-dispersion `noisy` dataset exhibit different execution dynamics compared to those trained on `play` (Fig. 3).

We now analyze these effects through the geometry of the learned trajectory energy.

A. Energy-Guided Execution Dynamics

NEaR performs planning by minimizing the learned trajectory energy under boundary constraints. At each replanning step, multiple candidate trajectories are generated by inpainting the goal state at different future indices t_g and refining them via gradient descent. The candidate with minimum energy is selected according to Eq. 12.

Fig. 4 visualizes the energy landscape over candidate goal indices at different execution steps. The selected trajectory consistently corresponds to the minimum-energy basin, demonstrating that inference operates as constrained optimization over an explicit objective.

Fig. 5 further shows that the selected trajectory exhibits progressively decreasing goal distance t_g as execution proceeds. This indicates that gradient-based refinement produces plans that become increasingly aligned with the task objective over time.

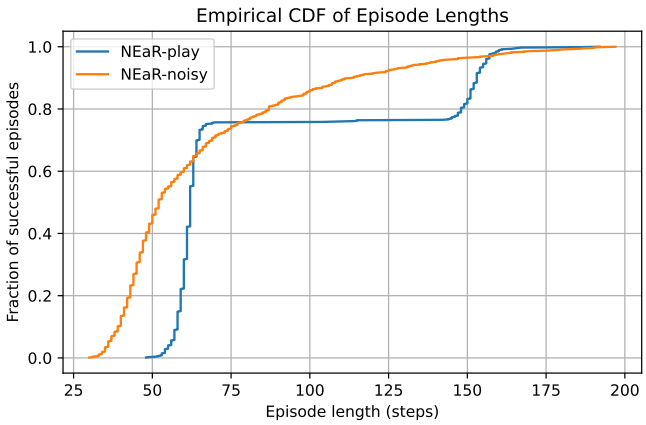


Fig. 3. **Empirical CDF of successful episode lengths for NEaR trained on `play` and `noisy` datasets.** The curves are computed from benchmark evaluation rollouts (not from the training data). NEaR trained on `noisy` demonstrations exhibits a smoother and earlier accumulation of successful completions, indicating that it often resolves the task at intermediate timesteps. In contrast, the `play`-trained model shows a bimodal pattern: many successes occur around $\sim$60 steps, while additional recoveries occur later ($\sim$160 steps), suggesting that when early refinement fails, the policy often requires a longer corrective sequence before succeeding.

Together, these diagnostics confirm that NEaR does not merely imitate local transitions; it performs structured optimization over trajectory space.

B. Effect of Dataset Dispersion on Optimization Geometry

The key difference between the `play` and `noisy` variants lies in state-space coverage. The `play` dataset contains smooth but narrowly distributed trajectories, whereas the `noisy` dataset introduces perturbations that expand coverage around feasible behaviors.

This difference is reflected in execution statistics (Fig. 3). The `noisy`-trained model shows a gradual accumulation of successful completions across a wide range of episode lengths, indicating flexible recovery and intermediate task resolution. In contrast, the `play`-trained model exhibits a bimodal completion pattern: many successes occur early, while others require substantially longer corrective sequences.

These dynamics are consistent with differences in the learned energy geometry. When trained on narrow data support, the energy landscape forms sharp valleys around demonstrated trajectories. Gradient information is therefore most reliable near the demonstration manifold, and larger deviations may require longer corrective refinement.

In contrast, broader dispersion encourages the model to learn gradients that map perturbed states back toward feasible trajectories. This effectively widens low-energy basins, making refinement more stable under distribution shift.

Importantly, this geometric interpretation aligns with the aggregate performance results (Fig. 2). Broader state coverage improves goal-reaching performance when an optimizable objective is available, even when demonstrations are individually suboptimal.

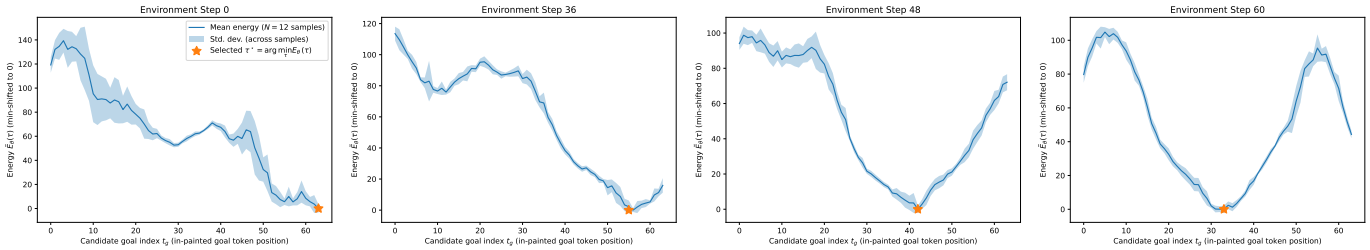


Fig. 4. **Energy over candidate goal indices.** At each replanning call, we in-paint the goal state at different candidate indices t_g and refine trajectories under boundary constraints. The curve shows mean and std. dev. of the learned trajectory energy $\tilde{E}_\theta(\tau)$ across N=12 stochastic samples per t_g (energies are min-shifted to 0 for visualization). NEaR selects the trajectory $\tau^\star$ with minimum energy (orange marker), i.e., maximum proxy reward $r(\tau) = -E_\theta(\tau)$. *Example rollout of NEaR learned on play data.

C. Implications

Taken together, these findings suggest that dataset dispersion influences not only data diversity but the geometry of the learned objective itself. When planning is formulated as constrained energy minimization, broader coverage improves the reliability of gradient-based refinement across a larger region of trajectory space.

NEaR isolates this mechanism by decoupling objective optimization from temporal-difference learning. The observed performance gains therefore arise from the geometry of the learned energy landscape rather than from value iteration or reward regression.

This perspective provides a minimal explanation for why higher-variance offline datasets can improve goal-conditioned performance when an explicit optimization objective is present.

VI. Conclusions

We introduced NEaR (Negative Energy as Reward), a reconstruction-based trajectory energy model for offline goal-conditioned control. NEaR is trained purely from demonstration data without reward labels or temporal-difference learning, yet enables goal-directed behavior through gradient-based refinement over a learned energy landscape.

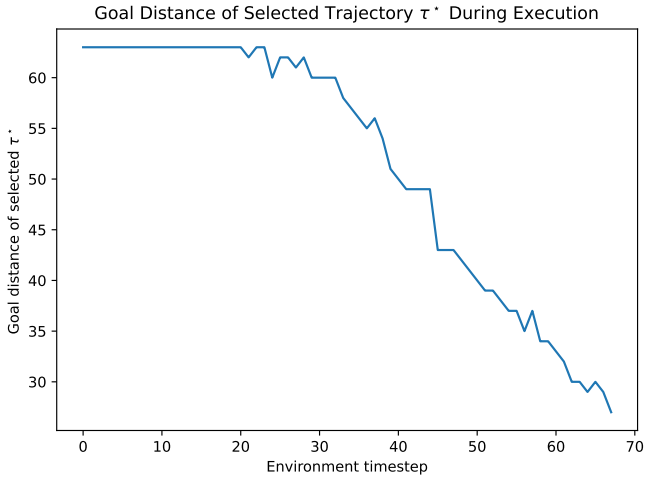


Fig. 5. At each environment timestep, NEaR selects the candidate trajectory $\tau^\star = \arg\min_\tau E_\theta(\tau)$. The plot shows the distance between the goal token index t_g of trajectory $\tau^\star$ as execution proceeds. Goal distance decreases almost monotonically, demonstrating that energy-guided refinement produces progressively more goal-aligned plans. Example empirical rollout.

Across the `cube-single` task, NEaR reproduces performance trends commonly observed in value-based methods: broader dataset dispersion is associated with improved goal-reaching performance when an optimizable objective is available. Diagnostic analyses show that inference operates as constrained energy minimization and that execution dynamics are shaped by the geometry of the learned energy landscape. These results suggest that part of the advantage attributed to reinforcement learning in offline settings may stem from the presence of an explicit objective that supports trajectory-level optimization.

The experimental study presented here is limited in scope and focuses on a single benchmark environment and state-based observations. As such, the findings should be interpreted as preliminary evidence rather than a comprehensive evaluation. Future work will extend this analysis to additional tasks, visual inputs, and larger-scale datasets, and will investigate theoretical properties of energy-based trajectory refinement, including stability and convergence under distribution shift. Exploring hybrid formulations that integrate energy-based objectives with online data collection is another promising direction.

Overall, this work provides an initial step toward understanding how optimization can complement imitation learning in reward-free offline control.

Reproducibility Statement. The full implementation of NEaR is publicly available at: https://github.com/cvg25/near. The repository includes training and inference code, hyperparameters, and evaluation scripts for OGBench.

References

[1] T. Osa, J. Pajarinen, G. Neumann, J. A. Bagnell, P. Abbeel, and J. Peters, "An algorithmic perspective on imitation learning," *Foundations and Trends® in Robotics*, vol. 7, no. 1-2, pp. 1–179, 2018.

[2] C. Lynch, M. Khansari, T. Xiao, V. Kumar, J. Tompson, S. Levine, and P. Sermanet, "Learning latent plans from play," in *Conference on robot learning*. Pmlr, 2020, pp. 1113–1132.

[3] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," *The International Journal of Robotics Research*, vol. 44, no. 10-11, pp. 1684–1704, 2025.

[4] N. M. Shafiullah, Z. Cui, A. A. Altanzaya, and L. Pinto, "Behavior transformers: Cloning k modes with one stone," *Advances in neural information processing systems*, vol. 35, pp. 22 955–22 968, 2022.
[5] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," *arXiv preprint arXiv:2304.13705*, 2023.
[6] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu *et al.*, "Rt-1: Robotics transformer for real-world control at scale," *arXiv preprint arXiv:2212.06817*, 2022.
[7] R. S. Sutton, A. G. Barto *et al.*, *Reinforcement learning: An introduction*. MIT press Cambridge, 1998, vol. 1, no. 1.
[8] S. Park, K. Frans, B. Eysenbach, and S. Levine, "Ogbench: Benchmarking offline goal-conditioned rl," *arXiv preprint arXiv:2410.20092*, 2024.
[9] V. Sobal, W. Zhang, K. Cho, R. Balestriero, T. G. Rudner, and Y. LeCun, "Learning from reward-free offline data: A case for planning with latent dynamics models," *arXiv preprint arXiv:2502.14819*, 2025.
[10] D. Brown, W. Goo, P. Nagarajan, and S. Niekum, "Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations," in *International conference on machine learning*. PMLR, 2019, pp. 783–792.
[11] D. S. Brown, W. Goo, and S. Niekum, "Better-than-demonstrator imitation learning via automatically-ranked demonstrations," in *Conference on robot learning*. PMLR, 2020, pp. 330–359.
[12] L. Chen, R. Paleja, and M. Gombolay, "Learning from suboptimal demonstration via self-supervised reward regression," in *Conference on robot learning*. PMLR, 2021, pp. 1262–1277.
[13] A. Kumar, A. Zhou, G. Tucker, and S. Levine, "Conservative q-learning for offline reinforcement learning," *Advances in neural information processing systems*, vol. 33, pp. 1179–1191, 2020.
[14] I. Kostrikov, A. Nair, and S. Levine, "Offline reinforcement learning with implicit q-learning," *arXiv preprint arXiv:2110.06169*, 2021.
[15] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, F. Huang *et al.*, "A tutorial on energy-based learning," *Predicting structured data*, vol. 1, no. 0, 2006.
[16] M. Janner, Y. Du, J. B. Tenenbaum, and S. Levine, "Planning with diffusion for flexible behavior synthesis," *arXiv preprint arXiv:2205.09991*, 2022.
[17] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," *arXiv preprint arXiv:2010.02502*, 2020.