Offline RLAIF: Piloting VLM Feedback for RL via SFO

Jacob Beck[1]

Keywords: RLAIF, RLHF, Feedback, Offline, RL, VLM

Summary

While internet-scale image and textual data have enabled strong generalization in Vision-Language Models (VLMs), the absence of internet-scale control data has impeded the development of similar generalization in standard reinforcement learning (RL) agents. Although VLMs are fundamentally limited in their ability to solve control tasks due to their lack of action-conditioned training data, their capacity for image understanding allows them to provide valuable feedback in RL tasks by recognizing successful outcomes. A key challenge in Reinforcement Learning from AI Feedback (RLAIF) is determining how best to integrate VLM-derived signals into the learning process. We explore this question in the context of offline RL and introduce a class of methods called **Sub-Trajectory Filtered Optimization (SFO)**. We identify three key insights. First, trajectory length plays a crucial role in offline RL, as full-trajectory preference learning exacerbates the *stitching problem*, necessitating the use of sub-trajectories. Second, even in Markovian environments, a non-Markovian reward signal from a sequence of images is required to assess trajectory improvement, as VLMs do not interpret control actions and must rely on visual cues over time. Third, a simple yet effective approach—filtered and weighted behavior cloning—consistently outperforms more complex RLHF-based methods. We propose **Sub-Trajectory Filtered Behavior Cloning (SFBC)**, a method that leverages VLM feedback on sub-trajectories while incorporating a *retrospective filtering* mechanism that removes sub-trajectories preceding failures to improve robustness.

Contribution(s)

1. We introduce Sub-Trajectory Filtered Behavior Cloning (SFBC), an offline RL algorithm that filters and weights sub-trajectories based on success probabilities obtained directly by querying a VLM. To support this method, we also introduce a retrospective filtering technique that removes sub-trajectories preceding failures.
   Context: Prior work typically distills VLMs into smaller models, uses LLMs to define reward functions in code, performs direct preference optimization, or applies contrastive models to estimate goal completion heuristically.
2. We provide a preliminary evaluation of SFBC by comparing it to alternative VLM-based methods and standard offline RL algorithms using ground-truth rewards in a continuous control setting. We also conduct several ablations to validate each component.
   Context: While limited to a single domain, our experiments demonstrate that there exists a standard control task where SFBC significantly outperforms more complex baselines.
3. We identify that preference learning on full trajectories can be uninformative in offline RL, whereas sub-trajectory supervision mitigates this issue and enables effective learning. We corroborate this insight with empirical evidence, showing that full-trajectory preference baselines degrade both success rate and return.
   Context: To our knowledge, our work is the first to identify how RLAIF can exacerbate the *stitching* problem in offline RL, and how sub-trajectory selection can resolve it.
4. We highlight the importance of non-Markovian feedback for RLAIF, even in Markovian environments, since VLMs must rely on visual cues over time to reason about actions they were not trained to interpret. We empirically demonstrate that prompting for non-Markovian feedback improves performance.
   Context: To our knowledge, ours is the first work to emphasize the utility of non-Markovian feedback, even in Markovian environments, when querying VLMs directly.

Offline RLAIF: Piloting VLM Feedback for RL via SFO

Jacob Beck[1]
jacob_beck@alumni.brown.edu

[1]**Independent; Now at Oracle** | https://github.com/jacooba/OfflineRLAIF

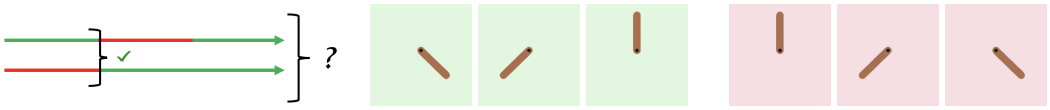
Abstract

While internet-scale image and textual data have enabled strong generalization in Vision-Language Models (VLMs), the absence of internet-scale control data has impeded the development of similar generalization in standard reinforcement learning (RL) agents. Although VLMs are fundamentally limited in their ability to solve control tasks due to their lack of action-conditioned training data, their capacity for image understanding allows them to provide valuable feedback in RL tasks by recognizing successful outcomes. A key challenge in Reinforcement Learning from AI Feedback (RLAIF) is determining how best to integrate VLM-derived signals into the learning process. We explore this question in the context of offline RL and introduce a class of methods called **Sub-Trajectory Filtered Optimization (SFO)**. We identify three key insights. First, trajectory length plays a crucial role in offline RL, as full-trajectory preference learning exacerbates the *stitching problem*, necessitating the use of sub-trajectories. Second, even in Markovian environments, a non-Markovian reward signal from a sequence of images is required to assess trajectory improvement, as VLMs do not interpret control actions and must rely on visual cues over time. Third, a simple yet effective approach—filtered and weighted behavior cloning—consistently outperforms more complex RLHF-based methods. We propose **Sub-Trajectory Filtered Behavior Cloning (SFBC)**, a method that leverages VLM feedback on sub-trajectories while incorporating a *retrospective filtering* mechanism that removes sub-trajectories preceding failures to improve robustness.

1 Introduction

A long-standing goal in reinforcement learning (RL) is to develop agents capable of solving a diverse set of tasks (i.e., broad generalization) while also learning efficiently from limited experience (i.e., rapid adaptation). Despite substantial progress, RL remains highly sample-inefficient and slow to adapt to new environments. Addressing these inefficiencies requires a foundation model for RL—one that generalizes across tasks and improves learning efficiency. However, a major obstacle to achieving this is the absence of large-scale, high-quality control datasets.

Reinforcement Learning from AI Feedback (RLAIF) presents a promising approach by leveraging vision-language models (VLMs) as scalable sources of learning signals (Bai et al., 2022; Klissarov et al., 2023; Ma et al., 2023; Faldor et al., 2024; Rocamonde et al., 2024; Baumli et al., 2023; Lee et al., 2024; Wang et al., 2024). However, determining how best to integrate VLM-derived signals into RL remains an open challenge. A key difficulty is that VLMs struggle to discriminate between similar trajectories, making it difficult to provide precise, informative feedback. This challenge is further complicated in online RL, where agents must learn from randomly initialized policies, leading to highly ambiguous early-stage feedback.



(a) Comparing full trajectories can be uninformative, whereas evaluating sub-trajectories can resolve ambiguities and better capture meaningful distinctions.

(b) An example trajectory where the agent successfully swings up a pendulum, which should receive positive feedback.

(c) A failure case where Markovian feedback assigns the same reward to a bad trajectory as a good one, failing to differentiate between them.

Figure 1: Illustration of the necessity of Non-Markov sub-trajectory feedback. (a) Full-trajectory comparisons can be misleading, while sub-trajectories allow more informative distinctions. (b) A successful trajectory of a pendulum swing-up task. (c) A failure case where Markovian feedback cannot distinguish between successful and unsuccessful trajectories.

Offline RL, in contrast, provides access to pre-collected data, which can include trajectories that are easier for VLMs to differentiate. For this reason, we focus on offline RL. However, integrating VLMs into offline RL introduces additional challenges, particularly in handling trajectory credit assignment and feedback propagation.

To address these issues, we introduce **Sub-Trajectory Filtered Optimization (SFO)**, a class of methods that effectively incorporate VLM-derived feedback to reach new heights in offline RL. Our study identifies three key insights:

1. First, trajectory length is a crucial factor in offline RLAIF, since full-trajectory evaluation exacerbates the well-known *stitching problem*. This problem necessitates the use of sub-trajectories, as depicted in Figure 1.
2. Second, even in Markovian environments, a non-Markovian reward model based on image sequences is required to assess trajectory improvement. Since VLMs are not trained on action-conditioned data, they have no basis for reasoning about how actions influence state transitions and must instead rely on visual cues over time. This is depicted in Figure 1. While discrete actions can sometimes be represented textually, many control tasks involve continuous actions (e.g., torques) that cannot be meaningfully encoded in a format suitable for VLMs. Furthermore, providing a reward at every time step by querying a VLM is computationally intractable for many problems without additional distillation into a smaller model (Wang et al., 2024), and sub-sampling states to estimate reward would already abandon the Markov property.
3. Third, despite the complexity of existing RLHF-based methods, we find that a simple yet effective approach—filtered and weighted behavior cloning—performs best in practice.

To that end, we introduce **Sub-Trajectory Filtered Behavior Cloning (SFBC)**, which leverages VLM feedback on sub-trajectories while incorporating a retrospective filtering mechanism that removes sub-trajectories preceding failures. We empirically evaluate SFBC in a toy control domain, demonstrating its effectiveness compared to existing baselines, including naive behavior cloning and preference-based methods such as Direct Preference Optimization (DPO) (Rafailov et al., 2023). Notably, our method outperforms AWAC (Nair et al., 2020) and TD3+BC (Fujimoto & Gu, 2021), even when those methods use ground-truth rewards, highlighting the effectiveness of VLM-driven sub-trajectory feedback.

2 Related Work

One approach to improving RL efficiency is the manual collection of large-scale offline datasets (Vuong et al., 2023). However, this is difficult to scale due to the expense of generating diverse and high-quality trajectories on robotics in the real world. Similarly, human annotation of RL datasets is costly and difficult to scale, given the need for expert-level feedback and the challenge of obtaining consistent reward signals across tasks.

VLMs offer a promising alternative to human annotation by providing AI-generated feedback. Prior work has explored using VLMs for reward design and environment generation in code. Eureka and OMNI-EPIC (Ma et al., 2023; Faldor et al., 2024) have proposed methods where VLMs write programs to specify rewards and structure environments for learning. However, just as machine learning is needed to specify object categories (e.g., recognizing a dishwasher) rather than relying on handcrafted rules, RL also requires machine-learned reward models rather than fixed rules from a VLM (e.g., to identify when a dishwasher has been unloaded). We firmly believe that hardcoded rules from a VLM cannot specify sufficiently complex tasks to elicit sufficiently robust reinforcement learning agents.

Recent work has also attempted to leverage AI-based reward detection to improve RLHF. Agent Q (Putta et al., 2024) uses AI-driven reward and success detection to generate preference labels for DPO. RL-VLM-F (Wang et al., 2024) distills a reward model from AI-generated labels but is an online method, and operates on Markovian states, which we find to be suboptimal. Rocamonde et al. (2024) and Baumli et al. (2023) use a CLIP model to provide Markovian feedback. Additionally, RoboCLIP (Sontakke et al., 2023) encodes text and video representations using CLIP to provide feedback. While RoboCLIP-style methods offer a useful way to encode feedback, we propose an end-to-end approach using a VLM (GPT-4o in our case), which provides greater flexibility. Our study focuses on how best to utilize this feedback and how to leverage feedback in an offline context.

Complementary approaches involve automatic environment generation and world modeling (Bruce et al., 2024; Valevski et al., 2024). These efforts provide a scalable way to generate diverse training environments, but they still require specifying reward signals to learn effectively. Similarly, automated task selection (Wang et al., 2023) and task ordering techniques such as Prioritized Level Replay (PLR) contribute to efficient learning but do not replace the need for automated high-quality feedback signals.

3 Methods

Our objective is to learn an optimal policy π^* in an offline RL setting using vision-language model (VLM) feedback. Given a dataset $\mathcal{D} = \{(s_t, a_t, s_{t+1}, a_{t+1}, \dots)\}$ of trajectories collected by unknown behavior policies, we seek to leverage VLMs to extract meaningful rewards and preferences over trajectory segments while mitigating the challenges of stitching with offline RL and Markovian rewards with VLMs, depicted in Figure 1.

3.1 Sub-Trajectory Filtered Behavior Cloning (SFBC)

To incorporate VLM feedback effectively, we propose **Sub-Trajectory Filtered Behavior Cloning (SFBC)**, which refines behavior cloning by selectively weighting and filtering trajectory segments based on VLM-derived success probabilities.

1. Sub-Trajectory Decomposition: Each trajectory $\tau = (s_1, a_1, s_2, a_2, \dots, s_T)$ is divided into n equal-length sub-trajectories τ_i, where:

$$\tau_i = (s_{i \cdot k}, a_{i \cdot k}, s_{i \cdot k+1}, a_{i \cdot k+1}, \dots, s_{(i+1) \cdot k}), \tag{1}$$

with segment length $k = \lfloor T/n \rfloor$. Additionally, we subsample observations within each segment to reduce token usage when querying the VLM.

2. VLM-Based Filtering: A VLM is queried with both a Markov prompt and a non-Markov prompt. For example, in the Pendulum-v1 environment, we use:

- **Markov:** "You are watching a video of a red stick. If the black dot is at the bottom of the stick, answer 'Y'. Otherwise, answer 'N'."
- **Non-Markov:** "You are watching a video of a red stick. If the stick has moved between sides of the screen (left to right or right to left), answer 'Y'. Otherwise, answer 'N'."

Each prompt is followed by a sequence of images representing the sub-trajectory. The probabilities of responses ("y") and ("yes") (after lower-casing the response) are summed, as are the probabilities of ("n") and ("no"). We qualitatively find that the collective "no" probability is more reliable, and both the Markov and non-Markov prompts contribute essential information. Thus, the final success probability is computed as:

$$P_{\text{Markov}}(\tau_i) = 1 - P(\text{"no"}|\text{Markov Prompt}), \tag{2}$$

$$P_{\text{Non-Markov}}(\tau_i) = 1 - P(\text{"no"}|\text{Non-Markov Prompt}), \tag{3}$$

$$P_{VLM}(\tau_i) = \min\left(1, P_{\text{Markov}}(\tau_i) + P_{\text{Non-Markov}}(\tau_i)\right). \tag{4}$$

A sub-trajectory is retained if $P_{VLM}(\tau_i) \geq \alpha$. To ensure robustness, we employ *retrospective* filtering, where a sub-trajectory τ_i is only included if the next sub-trajectory τ_{i+1} also meets the threshold:

$$\mathcal{D}_{SFBC} = \{(s_t, a_t, \tau_i) \mid \tau_i \in \mathcal{D},\ (s_t, a_t) \in \tau_i,\ PVLM(\tau_i) \geq \alpha,\ P_{VLM}(\tau_{i+1}) \geq \alpha\}. \tag{5}$$

This filtering mechanism assumes that a failed sub-trajectory likely resulted from preceding failures, ensuring that faulty transitions are not reinforced.

3. Weighted Behavior Cloning: The learned policy, π_θ, is then trained using the following behavior cloning objective:

$$\mathcal{L}_{SFBC} = -\mathbb{E}_{(s_t,a_t,\tau_i)\sim\mathcal{D}_{SFBC}}\left[P_{VLM}(\tau_i)\log\pi_\theta(a_t|s_t)\right]. \tag{6}$$

3.2 Comparison to Alternative SFO Methods

We contrast SFBC with alternative methods leveraging sub-trajectories, including:

- **VLM+TD3+BC:** Interprets $P_{VLM}(\tau)$ as a reward and applies TD3+BC (Fujimoto & Gu, 2021), a competitive offline RL algorithm.
- **S-DPO:** Applies Direct Preference Optimization (DPO) (Rafailov et al., 2023) to sub-trajectories using VLM-derived rankings.

4 Experiments

To evaluate the effectiveness of SFBC, we conduct experiments in the Pendulum-v1 environment, a standard continuous control benchmark. Our evaluation focuses on assessing SFBC's ability to correctly identify and stitch together the most optimal segments from sub-optimal demonstrations while outperforming conventional offline RL baselines. While the VLM (ChatGPT-4o) processes image-based observations to provide feedback, the learned policy itself conditions on lower-dimensional vector representations of the state, following standard RL formulations.

We compare SFBC against both standard offline RL algorithms and alternative VLM-assisted methods. To ensure robustness and statistical significance, we report success rate, mean return, and standard error across 15 seeds. Our analysis covers both overall performance comparisons and detailed ablation studies to isolate the contributions of each design choice in SFBC.

4.1 Dataset Construction

To evaluate the effectiveness of our proposed method in offline reinforcement learning, we construct a dataset for the Pendulum-v1 environment designed to explicitly test the ability to stitch optimal sub-trajectories together. Our dataset consists of 500 trajectories, each with a length of 600 time steps. These trajectories are generated from two distinct policies:

- **Expert Policy:** A custom proportional-derivative (PD) controller designed to stabilize the pendulum in an upright position. The controller balances the pendulum when near vertical and applies a predefined acceleration-based heuristic otherwise.

Method	**Success Rate (%)**	Std. Error (%)	**Mean Return**	Std. Error
BC Naive	33	12	-4716	790
TD3+BC (GT)	27	11	-5131	814
VLM BC (Full-Trajectory)	13	9	-5234	578
AWAC (GT)	0	0	-7840	308
SF-BC (Ours)	**73**	11	**-1585**	518
VLM+TD3+BC	27	11	-5013	649
S-DPO	0	0	-6859	181

Table 1: Performance comparison of SFBC against standard offline RL and VLM-assisted baselines in the Pendulum-v1 environment. SFBC achieves the highest success rate and mean return, demonstrating its effectiveness in leveraging VLM feedback for sub-trajectory optimization.

- **Failure Policy:** A manually constructed failure policy that consistently directs the pendulum downward, ensuring suboptimal performance.

To create a dataset that challenges the ability to stitch the best segments of given trajectories, each trajectory consists of one expert policy demonstration of length 300 and one failure policy demonstration of length 300, in a random order.

4.2 Baseline Comparisons

We compare our proposed method, SFBC, against a set of competitive baselines, including both standard offline RL algorithms and behavior cloning methods. Specifically, we evaluate:

- **BC Naive:** A standard behavior cloning baseline trained on the entire dataset, including both expert and failure trajectories, without any filtering or weighting mechanisms.
- **VLM BC (Full-Trajectory):** A VLM-assisted behavior cloning approach that applies success filtering at the full-trajectory level rather than sub-trajectories. Since our dataset contains a mix of expert and failure trajectories in random order, filtering at the full-trajectory level does not provide meaningful differentiation. We include this baseline primarily for didactic purposes.
- **TD3+BC (GT):** A strong offline RL baseline using TD3+BC (Fujimoto & Gu, 2021), trained with access to ground-truth rewards.
- **AWAC (GT):** Another standard offline RL algorithm, AWAC (Nair et al., 2020), trained on the dataset with ground-truth rewards.

The results, presented in Table 1, demonstrate that SFBC outperforms all baselines in both mean return and success rate. Notably, SFBC achieves better performance than TD3+BC trained on ground-truth rewards, underscoring the effectiveness of leveraging VLM-derived trajectory filtering and weighting. This suggests that VLM-derived feedback captures non-myopic task success, effectively bypassing the need for explicit reward propagation, a known challenge in offline RL due to overestimation biases. Additionally, SFBC surpasses S-DPO. We hypothesize that S-DPO fails due to inherent limitations in existing vision-language models, particularly their difficulty in reasoning over multiple sequences of visual data, which is essential for accurate trajectory ranking.

Furthermore, SFBC outperforms VLM+TD3+BC, justifying our choice to treat P_{VLM} as a weight rather than a reward. Conceptually, the weighting scheme can be interpreted as a value function Q^{π_θ}, such that the weighted loss, $\mathcal{L}_{SFBC}$, implicitly optimizes a policy gradient. However, $\mathcal{L}_{SFBC}$ lacks an importance sampling correction, and P_{VLM} is computed from offline data rather than the current policy, making it a better model of $Q^{\pi_{\text{behavior}}}$ (or $Q^{\pi_{\text{expert}}}$ after filtering). SFBC also resembles AWAC (Nair et al., 2020), except that the weighting P_{VLM} should likewise be interpreted as approximating $e^{A^{\pi_{\text{expert}}}}$, rather than the advantage of the learned policy $e^{A^{\pi_\theta}}$. While SFBC does not fully correspond

Ablation	**Success Rate (%)**	Std. Error (%)	**Mean Return**	Std. Error
SF-BC (Ours)	**73**	11	**-1585**	**518**
No Filtering	40	13	-4164	883
Markov Prompt Only	40	13	-4229	869
No Weighting	33	12	-3459	604
No Retrospective Filtering	13	9	-5562	525

Table 2: Ablation study on SFBC, evaluating the impact of key design choices. Removing filtering, non-Markovian feedback, weighting, or retrospective filtering significantly degrades performance, demonstrating the necessity of each component.

to a policy gradient, our results show that avoiding a direct reward interpretation—and the instability it introduces in reward propagation—is beneficial in practice.

4.3 Ablation Studies

We conduct ablation studies to assess the necessity of each component in SFBC. The results, presented in Table 2, confirm that each design choice contributes significantly to performance:

- **No Filtering:** Removing the filtering mechanism results in significantly worse performance, demonstrating the necessity of discarding low-confidence sub-trajectories.
- **Markov Prompt Only:** Using only the Markov prompt leads to degraded performance, confirming the importance of a non-Markovian understanding of trajectory improvement, even in Markovian environments, due to the inability of VLMs to process control data.
- **No Weighting:** Without weighting, the method lacks prioritization of high-quality sub-trajectories, leading to reduced performance.
- **No Retrospective Filtering:** This ablation resulted in the largest performance degradation. Without retrospective filtering, failure states remain in the dataset, potentially reinforcing poor decisions and leading to compounding errors in deployed policies. This highlights the importance of discarding preceding sub-trajectories that contribute to unsuccessful outcomes.

The ablations validate our hypothesis that all proposed modifications are necessary, further reinforcing the importance of VLM-informed filtering and weighting in offline RL.

5 Conclusion

In this work, we introduced Sub-Trajectory Filtered Behavior Cloning (SFBC), a method that leverages vision-language model (VLM) feedback for offline reinforcement learning by selectively filtering and weighting sub-trajectories. Our results demonstrate that SFBC outperforms standard offline RL methods such as TD3+BC and AWAC, as well as VLM-assisted baselines, validating the effectiveness of leveraging VLM-derived success probabilities for policy learning.

Our study provides key insights into the integration of VLM feedback with offline RL. First, sub-trajectory selection is critical, as full-trajectory preference learning exacerbates the stitching problem, making effective credit assignment impossible. Using sub-trajectories allows for improved learning stability and the ability to extract high-quality behaviors from mixed datasets. Second, non-Markovian feedback is essential, even in Markovian environments, since VLMs lack an understanding of control dynamics. Instead, they require vision-based trajectory analysis to correctly assess improvement over time. Our results confirm that non-Markovian prompts lead to significantly better performance. Finally, SFBC consistently outperforms competitive baselines, demonstrating that treating VLM-derived success probabilities as weights rather than rewards leads to superior offline policy learning. Our retrospective filtering mechanism further enhances performance by re-

moving failure-propagating sub-trajectories. Our findings suggest that VLM feedback provides non-myopic guidance, mitigating the need for explicit reward propagation and addressing key challenges in scaling offline RL.

Most immediately, this work could be applied, at scale, to automate feedback across a sufficiently diverse set of tasks to support training a general RL foundation model from offline data. Training the RL agent to be adaptive would inherently fall under the category of offline meta-RL (Beck et al., 2023; 2025). Moreover, implementing the agent with a general-purpose sequence model, such as a transformer (Vaswani et al., 2017), would also place it within the domain of in-context RL (Moeini et al., 2025), while also obviating the need for complex meta-gradient computation (Vuorio et al., 2021). Putting feedback and guidance systems on autopilot may finally provide the runway needed to arrive at a truly generalist agent for RL.

Another avenue for future research is exploring how offline feedback methods like SFBC can be extended to online RL. Since offline algorithms can be iteratively applied in an online setting, they provide a natural foundation for developing online reinforcement learning frameworks. Finally, while we leverage a vision-language model (VLM) trained primarily for static images, future work could explore the use of models specifically trained for video understanding, which may provide more accurate and temporally consistent trajectory assessments. As open-source and state-of-the-art VLMs continue to improve in video understanding, their ability to assess trajectory quality will become increasingly crucial for reinforcement learning applications to take off.

References

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.

Kate Baumli, Satinder Baveja, Feryal Behbahani, Harris Chan, Gheorghe Comanici, Sebastian Flennerhag, Maxime Gazeau, Kristian Holsheimer, Dan Horgan, Michael Laskin, Clare Lyle, Hussain Masoom, Kay McKinney, Volodymyr Mnih, Alexander Neitz, Dmitry Nikulin, Fabio Pardo, Jack Parker-Holder, John Quan, Tim Rocktäschel, Himanshu Sahni, Tom Schaul, Yannick Schroecker, Stephen Spencer, Richie Steigerwald, Luyu Wang, and Lei Zhang. Vision-language models as a source of rewards. *arXiv preprint arXiv:2312.09187*, 2023.

Jacob Beck, Risto Vuorio, Evan Zheran Liu, Zheng Xiong, Luisa Zintgraf, Chelsea Finn, and Shimon Whiteson. A survey of meta-reinforcement learning. *arXiv preprint arXiv:2301.08028*, 2023.

Jacob Beck, Risto Vuorio, Evan Zheran Liu, Zheng Xiong, Luisa Zintgraf, Chelsea Finn, and Shimon Whiteson. A tutorial on meta-reinforcement learning. *Foundations and Trends in Machine Learning*, 2025.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024.

Maxence Faldor, Jenny Zhang, Antoine Cully, and Jeff Clune. Omni-epic: Open-endedness via models of human notions of interestingness with environments programmed in code. *arXiv preprint arXiv:2405.15568*, 2024.

Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems*, 34:20132–20145, 2021.

Martin Klissarov, Pierluca D'Oro, Shagun Sodhani, Roberta Raileanu, Pierre-Luc Bacon, Pascal Vincent, Amy Zhang, and Mikael Henaff. Motif: Intrinsic motivation from artificial intelligence feedback. *arXiv preprint arXiv:2310.00166*, 2023.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. Rlaif vs. rlhf: Scaling reinforcement learning from human feedback with ai feedback. *ICML 2024*, 2024.

Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Eureka: Human-level reward design via coding large language models. *arXiv preprint arXiv:2310.12931*, 2023.

Amir Moeini, Jiuqi Wang, Jacob Beck, Ethan Blaser, Shimon Whiteson, Rohan Chandra, and Shangtong Zhang. A survey of in-context reinforcement learning. *arXiv*, 2025.

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

Pranav Putta, Edmund Mills, Naman Garg, Sumeet Motwani, Chelsea Finn, Divyansh Garg, and Rafael Rafailov. Agent q: Advanced reasoning and learning for autonomous ai agents. *arXiv preprint arXiv:2408.07199*, 2024.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741, 2023.

Juan Rocamonde, Victoriano Montesinos, Elvis Nava, Ethan Perez, and David Lindner. Vision-language models are zero-shot reward models for reinforcement learning. *International Conference on Learning Representations (ICLR) 2024*, 2024.

Sumedh Sontakke, Jesse Zhang, Séb Arnold, Karl Pertsch, Erdem Bıyık, Dorsa Sadigh, Chelsea Finn, and Laurent Itti. Roboclip: One demonstration is enough to learn robot policies. *Advances in Neural Information Processing Systems*, 36:55681–55693, 2023.

Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion models are real-time game engines. *arXiv preprint arXiv:2408.14837*, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 2017.

Quan Vuong, Sergey Levine, Homer Rich Walke, Karl Pertsch, Anikait Singh, Ria Doshi, Charles Xu, Jianlan Luo, Liam Tan, Dhruv Shah, et al. Open x-embodiment: Robotic learning datasets and rt-x models. In *Towards Generalist Robots: Learning Paradigms for Scalable Skill Acquisition@ CoRL2023*, 2023.

Risto Vuorio, Jacob Austin Beck, Gregory Farquhar, Jakob Nicolaus Foerster, and Shimon Whiteson. No dice: An investigation of the bias-variance tradeoff in meta-gradients. In *Deep RL Workshop NeurIPS 2021*, 2021.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

Yufei Wang, Zhanyi Sun, Jesse Zhang, Zhou Xian, Erdem Biyik, David Held, and Zackory Erickson. Rl-vlm-f: Reinforcement learning from vision language foundation model feedback. *arXiv preprint arXiv:2402.03681*, 2024.

Supplementary Materials

The following content was not necessarily subject to peer review.

A Additional Experiments and Hyperparameters

A.1 Additional VLM+TD3+BC Experiments

We tested a variation where filtering was applied before TD3+BC, but it did not significantly improve performance. We also tested VLM+TD3+BC with the "Markov prompt only" to ensure rewards for TD3 were Markovian, as well as with a discount factor of $\gamma = 0.5$ to encourage myopic behavior. Neither approach led to improved performance. The results are summarized in Table 3.

Method	Return Mean	Std Err	Success Rate (%)
VLM+TD3+BC (Standard)	-5013	649	27
VLM+TD3+BC (Filtered Before TD3+BC)	-4576	660	27
VLM+TD3+BC (Markov Prompt Only)	-4727	705	27
VLM+TD3+BC ($\gamma = 0.5$)	-5048	523	20

Table 3: Performance comparison of VLM+TD3+BC variants. Filtering before TD3+BC, using a Markov-only prompt, or reducing γ did not significantly improve performance.

A.2 Prompting Details

For DPO, we used the following prompt:

> "You are watching two videos of a red stick. The goal is to swing the stick up to gain height. It is bad to let it fall and lose momentum. Respond '1' if Video 1 is better, '2' if Video 2 is better."

Followed by:

> "Video 1:" <sub-trajectory>
> "Video 2:" <sub-trajectory>

A.3 Hyperparameters

We used the default hyperparameters from D3RLpy for all baselines. Additionally, we tested a learning rate of $1e^{-3}$ for AWAC and TD3+BC, as this is the default for BC, but it did not improve results. We found $\alpha = 0.1$ to be the most effective threshold for filtering.

For sub-trajectory processing, we used:

- Sub-trajectory length: 100
- Subsampling factor: 20

Success rates, mean returns, and standard errors are reported across 15 seeds to ensure statistical robustness.