Consensus Sampling in Contrastive Decoding for Imitation Learning in Robotics

Rhea Malhotra[1]
rheamal@stanford.edu
Mentor: Yuejiang Liu & Chelsea Finn
[1]Department of Computer Science, Stanford University

June 11, 2024

See Code Here! See Demo Here!

Abstract

Here, we leverage inter-sample relationships in demonstration quality using consensus sampling for contrastive imitation learning for robotics applications. By implementing mechanisms such as consensus among neighboring samples within contrastive decoding between ranked demonstrations, we aim to improve robust online human behavioral cloning for robot learning. Our model, based on that of Consistency Policy outperforms baseline behavior cloning approaches in both vanilla and consensus selected behavior cloning. We successfully integrated consensus sampling with contrastive learning for behavior cloning, demonstrating the viability of our approach for annotating human demonstrations. To enhance robustness to handle periodic variance in rewards, we add temporal noise to maintain performance despite temporal correlation. Experiments showed that binary and continuous scoring methods yield similar performance in the PushT task, with a final 92.4% success rate in binary contrastive imitation learning. Future research directions include pairwise mapping of similar states to good-bad demonstrations, expanding the approach to more tasks, and implementing online reinforcement learning.

1 Introduction

Learning from human demonstrations in dynamic environments poses substantial challenges, particularly in robotics, where sampling inaccuracies can lead to suboptimal or dangerous actions. Practical challenges stemming from the non-determinism of human demonstrations pose challenges to implementation and sampling in action rollout. Not all sampled sequences are consistent and equally optimal and random sampling can lead to erratic and even dangerous outcomes. Here, I seek to develop a behavioral cloning approach that leverages inter-sample relationships to learn human demonstrations. By implementing mechanisms such as consensus among neighboring samples within contrastive decoding between ranked demos, we aim to experiment with methods to improve robust online human behavioral cloning for robot learning.

2 Related Work

This work is primarily inspired by the works of Contrastive Decoding for open-ended text generation in the field of NLP [Li et al.(2022)Li, Holtzman, Fried, Liang, Eisner, Hashimoto, Zettlemoyer, and Lewis]. Language models (LM) present the challenge of generating random and diverse yet accurate output, where greedy decision making and maximum probability is a poor decoding objective. In the context of imitation learning, similar greedy approaches would produces short and repetitive

sequence trajectories. Contrastive decoding (CD) presents a contrastive objective subject to a plausibility constraint, which returns the difference between and expert and amateur likelihood. We use the inspiration of this work to define the loss function and objective framework by which we train on 'good' and 'bad' human demonstrations, as equivalents to an expert and amateur model, respectively.

Further, we use Consensus Sampling as a means of distinguishing quality of human demonstration. Sekhari, et. al proposed selective sampling to actively query the noisy expert for feedback. Their selective sampling algorithm works with general function classes and multiple actions and provides a framework by which to integrate noisy expert for robustness [Sekhari et al.(2024)Sekhari, Sridharan, Sun, and Wu]. This follows inline with the concepts of Divergence Minimization or the ideas of bootstrapping from expert demonstrations. A challenge in the field of behavior cloning is that policies fail due to limited demonstrations datasets, a scenario in which BC methods often fail. The paper shows f-MAX, an f-divergence generalization of AIRL for IRL's state-marginal matching objective contributes most to its superior performance. [Ghasemipour et al.(2020)Ghasemipour, Zemel, and Gu].

[Ma et al.(2023)Ma, Hu, Wang, and Sun] [Bertsch et al.(2023)Bertsch, Xie, Neubig, and Gormley]

Imitation Learning

Diffusion Policy paved the way for generating robot behavior by representing a robot's visuomotor policy as a conditional denoising diffusion process. We use these foundational findings and robot manipulation benchmarks with improvement relative to their given baseline of 46.9%. Diffusion Policy learns the gradient of the action-distribution score function and iteratively optimizes with respect to this gradient field, with receding horizon control, visual conditioning, and the time-series diffusion transformer. These learning techniques inspire our implementation of contrastive decoding in imitation learning, with defined sucess metrics and dataset modules. [Chi et al.(2023)Chi, Feng, Du, Xu, Cousineau, Burchfiel, and Song]

We primarily build upon the work of the Consistency Policy, 2024 paper which directly builds upon Diffusion policy [Prasad et al.(2024)Prasad, Lin, Wu, Zhou, and Bohg], to address the high-end GPU contraints in achieving fast policy inference in robotic applications. Consistency Policy uses a pretrained Diffusion Policy by enforcing self-consistency along the Diffusion Policy's learned trajectories. Specifically, we also use this paper's demonstrations module for the PushT dataset. [Wang et al.(2022)Wang, Wei, Schuurmans, Le, Chi, Narang, Chowdhery, and Zhou]

3 Approach

3.1 Motivation: Lack of Annotation in Human Demonstration Quality

A persistent challenge in imitation learning is effectively leveraging the variance in quality of human demonstration data, especially when such data is not annotated with preference labels indicating whether a demonstration is good or bad. This lack of annotation complicates the implementation of contrastive decoding, which relies on distinguishing between good and bad demonstrations for effective behavior cloning. The core problem lies in the sampling and annotation process for contrastive decoding in human demonstrations. Traditional offline reinforcement learning (RL) techniques assume a reward structure closely tied to the demonstrations, but without explicit quality annotations, this assumption becomes non-trivial to maintain.

Here, under the assumption that most human demonstrations are successful and suboptimal demonstrations are considered outliers in unimodal demonstration tasks, we develop a sampling method to assess the quality of demonstration data. In cases where pretrained visual language models cannot provide automatic labeling, we utilize consensus sampling to assume that successful behaviors follow neighboring trajectories, thereby distinguishing between good and bad demonstrations.

By exploring weak priors and consensus-based sampling techniques, we assume that most

human demonstrations are successful. Leveraging relationships between samples allows us to mitigate the lack of explicit annotations and still achieve effective behavior cloning. Implementing consensus-based sampling involves drawing multiple sequences where most are expected to be good based on weak priors. Additionally, our sampling method introduces noise into demonstrations to capture temporal correlations and evaluate the robustness of the method. Finally, this project uses diffusion policy techniques to explore the effectiveness and limitations of the sampling algorithm.

3.2 Datasets for Simulation Experiments

To implement sampling and Contrastive Decoding, we use the human demonstration datasets provided from Consistency Policy paper, we conducted experiments across six tasks using three established benchmarks: Robomimic, Push-T, and Franka Kitchen, standard in visuomotor and state-based policy learning [Chi et al.(2023)Chi, Feng, Du, Xu, Cousineau, Burchfiel, and Song] and ParaDiGMS . Here, we focus on the PushT task.

Initially, we attempted using datasets from the (RH20T Website) and a Lerobot repository environment, which ultimately was also too complex for the purposes of consensus sampling. Finally, we settled on the data from the contrastive decoding paper because The simulation setup from consistency and diffusion policy papers for faster, low-latency evaluations in a restricted GPU environment.

The Push-T task involves pushing a T-shaped block to a fixed target using a circular end-effector. We used a dataset of 200 expert demonstrations from [Chi et al.(2023)Chi, Feng, Du, Xu, Cousineau, Bu and report results for policies using state-based observations. We evaluate the policy every 50 epochs with the success rate logged on wandb, with rollout videos as well.

Figure 1: Push-T task rollout in Lerobot Simulation.

3.3 Sampling Method

To enhance online robotic behavior cloning, we implement a contrastive decoding approach that leverages consensus among neighboring samples within a batch. This approach classifies samples as 'good' or 'bad' based on their proximity to optimal behaviors.

Consensus Sampling: In each batch, samples that are close to neighboring trajectories, as measured by vector distance to trajectory similarity, are considered 'good'. This proximity indicates a high probability of successful behavior. Conversely, samples that are far from these neighboring trajectories are labeled as 'bad'. These samples are weighted by their lower probability distribution and vector distance from the 'good' trajectory, indicating suboptimal behaviors. To further refine this approach, distinctions are made between samples from a pair of weak and strong models. This comparison helps reduce biases by highlighting the differences in sample quality between the two models.

Define a score for each demonstration based on a desired metric for the PushT task. We consider the percentage of coverage of the T symbol as well as the degrees of alignment with the baseline T orientation, with a higher score indicating closer proximity to the final desired position:

$$\text{score}(s, a) = \text{Coverage}(s, a) + \left(1 - \frac{\text{Alignment}(s, a)}{180}\right)$$

Calculate Mean and Standard Deviation of Scores

Calculate the mean (μ) and standard deviation (σ) of scores across all demonstrations:

$$\mu = \frac{1}{|D|} \sum_{(s,a) \in D} \text{score}(s, a)$$

$$\sigma = \sqrt{\frac{1}{|D|} \sum_{(s,a) \in D} (\text{score}(s, a) - \mu)^2}$$

Selecting Demonstrations Based on Consensus

Select demonstrations that exceed the mean score adjusted by a factor of the standard deviation:

$$D_{\text{consensus}} = \{(s, a) \in D \mid \text{score}(s, a) > \mu + k\sigma\}$$

Sampling from Consensus

Sample from $D_{\text{consensus}}$ to train the model, focusing on the most representative examples of desired behavior. This was fit manually across samples. Additionally, we incorporate a technique that uses weighted average centroid markers of good demonstrations. This method helps in creating a gradient of demonstration strength rather than a binary classification. Demonstrations far from the centroid of good samples are assigned lower weights, indicating suboptimal performance. This creates a metric of demonstration strength, providing a nuanced classification instead of a simple binary label.

This sampling method aims to sample behaviors with desired properties during deployment, eventually without relying on explicit preference annotations. By focusing on the quality of demonstrations and leveraging weak priors, the method ensures that the robot learns robust and effective behaviors even in the absence of direct labels.

3.4 Noise for Robustness

To enhance the robustness of our learning algorithm and ensure that it can handle real-world variations, we introduce noise into the demonstration data. In sampling, first we establish a probability, p, with which noise will be introduced into the demonstrations. This simulates realistic errors or variations in human demonstrations. Then, we ensure that the noise is not entirely random but has temporal correlation. Noise persists for a certain duration, creating a locally correlated noise pattern over time. For each demonstration, noise is introduced according to the defined probability This forces the learner to understand and adapt to variations over time, which is critical for developing robust policies. Simulating sub-optimality via variance further ensures that our learning algorithm can handle practical, noisy data. The presence of noise also helps in identifying outliers and improving the algorithm's ability to distinguish between good and bad demonstrations.

3.5 Contrastive Decoding Implementation

In contrastive imitation learning, the objective function can be represented as follows:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp(f_\theta(s_i,a_i))}{\exp(f_\theta(s_i,a_i))+\sum_{j=1,j\neq i}^{N}\exp(f_\theta(s_i,a_j))}$$

where s_i represents the state, a_i represents the action, and $f_\theta(s_i,a_i)$ is the feature representation of the state-action pair (s_i,a_i) obtained from a neural network parameterized by θ.

In the binary case, each demonstration is labeled as either good or bad. This is simpler to handle, as the decision is categorical. Let σ(s,a) be a binary indicator function that outputs 1 for good demonstrations and 0 for bad demonstrations. This loss function uses the binary labels directly to push the model towards preferring actions labeled as good and away from actions labeled as bad. The contrastive loss function maximizes the distance between good and bad demonstrations:

$$L(\theta) = -\left[\sum_{(s,a)\in D}\delta(s,a)\log\frac{\exp(f_\theta(s,a))}{\sum_{a'\in\mathcal{A}}\exp(f_\theta(s,a'))}+(1-\delta(s,a))\log\frac{\exp(-f_\theta(s,a))}{\sum_{a'\in\mathcal{A}}\exp(-f_\theta(s,a'))}\right]$$

When scores are continuous, they provide a measure of the quality of a demonstration, normalized and ranging from a minimum to a maximum value. We continue to define score(s,a) as a function of coverage and orientation in the PushT task which returns a continuous value representing the quality of the demonstration. We normalize the scores to ensure they are bounded between 0 and 1 (which can be interpreted as probabilities or confidence levels of demonstrations being good):

$$\text{score}_{\text{norm}}(s,a) = \frac{\text{score}(s,a)-\min(\text{score})}{\max(\text{score})-\min(\text{score})}$$

This approach allows for a more nuanced differentiation between demonstrations, with the model learning more strongly from those with higher scores. The loss function can then weight contrastive terms by these normalized scores:

$$L(\theta) = -\left[\sum_{(s,a)\in D}\text{score}_{\text{norm}}(s,a)\log\frac{\exp(f_\theta(s,a))}{\sum_{a'\in\mathcal{A}}\exp(f_\theta(s,a'))}+(1-\text{score}_{\text{norm}}(s,a))\log\frac{\exp(-f_\theta(s,a))}{\sum_{a'\in\mathcal{A}}\exp(-f_\theta(s,a'))}\right]$$

In both cases, these loss functions can be integrated into the training of an imitation learning model. For binary scores, the model learns a clear distinction between good and bad. For continuous scores, the model's updates are weighted by the degree to which a demonstration is considered good or bad, allowing for finer adjustments based on demonstration quality. These frameworks can be applied using standard optimization techniques in machine learning, adapting the parameters to minimize the loss function over progressive batches of training data.

4 Experiment Results

We compared continuous scoring to binary scoring in contrastive learning. The results demonstrated that continuous scoring provided more nuanced feedback, improving the model's ability to differentiate between varying levels of demonstration quality.

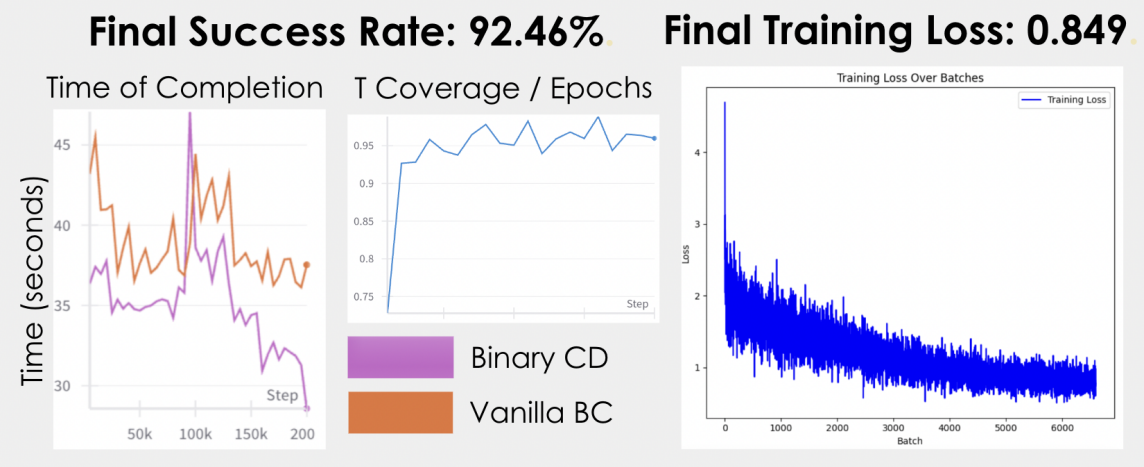

Figure 2: Results summary for Binary Contrastive Decoding versus Vanilla Behavior Cloning & Training Loss.

4.1 Consensus Strength Weighting

Manual tuning of the hyperparameter K was performed to determine the consensus strength weighting of the consensus threshold. The majority of human demonstrations were assumed to be successful. The value of K was fit based on the number of standard deviations to consider within the outliers' range. During the manual fitting process, setting K to values ≤ 1.5 resulted in the inclusion of suboptimal demos. On the other hand, setting K higher than 2.5 led to the exclusion of too many demonstrations, reducing the amount of useful training data and naturally resulting in overfitting. The selected value of $K = 1.6$ provided a balance, filtering out the majority of outliers while retaining a robust set of high-quality demonstrations for training. This choice of K allowed the model to achieve a better generalization performance compared to the baselines.

4.2 Hyperparameter Fitting

L2 regularization was integrated to counter overfitting, leading to more generalized model performance. After extensive testing, 500 epochs were found to provide an optimal balance between model complexity and true deterministic generation.

4.3 Baseline Comparisons

The final success rate achieved by our model was 92.46%. The final training loss was 0.849, indicating strong performance and efficient learning from the provided demonstrations. Performance is evaluated by measuring metrics such as success time and reward rates. To provide a comprehensive evaluation, we compared our contrastive decoding policy against two baselines: vanilla behavior cloning and behavior cloning using only the positive 'good' consensus samples.

Vanilla Behavior Cloning: This baseline involves training the model on all available demonstrations without differentiating between good and bad samples. While it provides a basic understanding of the overall demonstration quality, it does not leverage the potential benefits of distinguishing between different demonstration qualities. The vanilla behavior cloning approach resulted in a success rate of 88.32%, with a training loss of 1.256. This lower success rate and higher training loss indicate that the model struggled to generalize effectively from the mixed-quality data.

Behavior Cloning of Positive 'Good' Consensus Samples: In this baseline, the model was trained exclusively on the demonstrations identified as 'good' by the consensus sampling method. By focusing on high-quality demonstrations, this approach aimed to improve the

learning efficiency and performance of the model. The behavior cloning of good consensus samples achieved a success rate of 85.74% and a training loss of 1.024. This improvement over the vanilla behavior cloning demonstrates the benefit of leveraging high-quality data for training.

Contrastive Decoding Policy: Our proposed contrastive decoding policy outperformed both baselines, achieving a success rate of 92.46% and a training loss of 0.849, resulting in the optimal policy for this task.

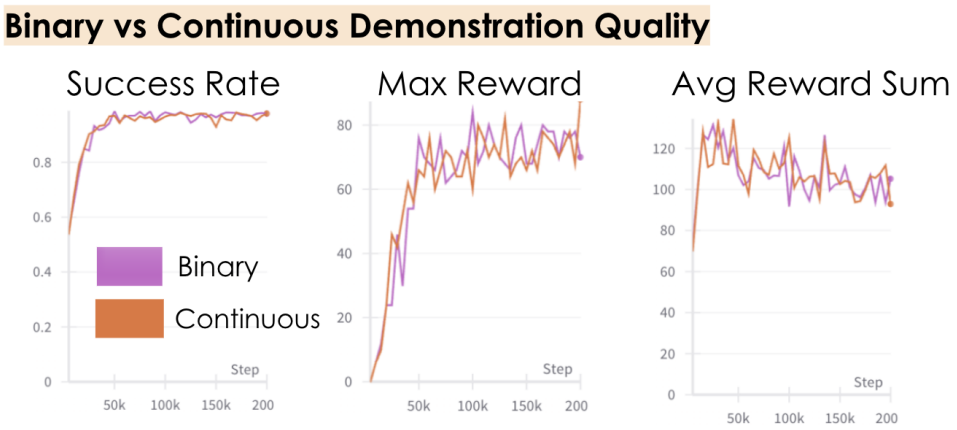

Figure 3: Comparing Continuous versus Binary scores in Contrastive Decoding Policy for Push-T.

5 Conclusion

Ultimately the goal of this project was to leverage inter-sample relationships in demonstration quality using consensus sampling for contrastive imitation learning. Our findings indicate that our model outperforms baseline behavior cloning approaches. We successfully integrated consensus sampling with contrastive learning for behavior cloning, demonstrating the viability of our approach for annotating human demonstrations. To enhance robustness, we incorporated noise to handle periodic variance in rewards. This approach effectively maintained performance despite temporal correlations. Our experiments showed that binary and continuous scoring methods yield similar performance in the PushT task.

Future research directions include pairwise mapping of similar states to good and bad demonstrations. Furthermore, we can expanding the approach to more tasks beyond PushT and implementing online reinforcement learning to further improve model performance and adaptability. Given the advancements in NLP, contrastive decoding alongside inter-sample consistency policy techniques is a promising technique for introducing unique behaviors in trajectory sequence generation for robotics.

References

[Bertsch et al.(2023)Bertsch, Xie, Neubig, and Gormley] Amanda Bertsch, Alex Xie, Graham Neubig, and Matthew R Gormley. 2023. It's mbr all the way down: Modern generation techniques through the lens of minimum bayes risk. *arXiv preprint arXiv:2310.01387*.

[Chi et al.(2023)Chi, Feng, Du, Xu, Cousineau, Burchfiel, and Song] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. 2023. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*.

[Ghasemipour et al.(2020)Ghasemipour, Zemel, and Gu] Seyed Kamyar Seyed Ghasemipour, Richard Zemel, and Shixiang Gu. 2020. A divergence minimization perspective on imitation learning methods. In *Conference on robot learning*, pages 1259–1277. PMLR.

[Li et al.(2022)Li, Holtzman, Fried, Liang, Eisner, Hashimoto, Zettlemoyer, and Lewis] Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2022. Contrastive decoding: Open-ended text generation as optimization. *arXiv preprint arXiv:2210.15097*.

[Ma et al.(2023)Ma, Hu, Wang, and Sun] Jiajun Ma, Tianyang Hu, Wenjia Wang, and Jiacheng Sun. 2023. Elucidating the design space of classifier-guided diffusion generation. *arXiv preprint arXiv:2310.11311*.

[Prasad et al.(2024)Prasad, Lin, Wu, Zhou, and Bohg] Aaditya Prasad, Kevin Lin, Jimmy Wu, Linqi Zhou, and Jeannette Bohg. 2024. Consistency policy: Accelerated visuomotor policies via consistency distillation. *arXiv preprint arXiv:2405.07503*.

[Sekhari et al.(2024)Sekhari, Sridharan, Sun, and Wu] Ayush Sekhari, Karthik Sridharan, Wen Sun, and Runzhe Wu. 2024. Selective sampling and imitation learning via online regression. *Advances in Neural Information Processing Systems*, 36.

[Wang et al.(2022)Wang, Wei, Schuurmans, Le, Chi, Narang, Chowdhery, and Zhou] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.