Appendix for *Out-of-Dynamics Imitation Learning from Multimodal Demonstrations*

Yiwen Qiu[1], **Jialong Wu**[2], **Zhangjie Cao**[3], **Mingsheng Long**[2]✉
[1]Department of Automation, Tsinghua University, China
[2]School of Software, BNRist, Tsinghua University, China
[3]Department of Computer Science, Stanford University, Stanford, CA 94305, USA
{qywmei,wujialong0229}@gmail.com
caozj@cs.stanford.edu, mingsheng@tsinghua.edu.cn

A Contrastive Clustering Algorithm

With the objectives introduced in the main text, we show our full contrastive clustering algorithm in Algorithm 1.

Algorithm 1: Contrastive Clustering Algorithm

```
Input: Demonstrations $\Xi$, Feature extractor $F$, Cluster center Matrix $\mathbf{C}$, Sub-trajectory length $l$,
       Learning rate $\alpha$ and $\lambda$
Initialize the parameters of $F$.
Randomly sample $K$ indices $j_k|_{k=1}^{K}$ from the interval $[1, N]$
Take a sub-trajectory $\xi_{j_k}^{\text{sub}}$ of length $l$ from $\xi_{j_k}$ and initialize $c_k$ with $F(\xi_{j_k}^{\text{sub}})$
while not converging do
    Sample $N$ trajectories $\{\xi_n\}_{n=1}^{N}$ from $\Xi$ and subsample two sub-trajectories $\xi_{2n-1}^{\text{sub}}$ and $\xi_{2n}^{\text{sub}}$ of
      length $l$ for each $\xi_n \in \Xi$.
    Assign the cluster label $\mathbf{y}_n$ to $\xi_n^{\text{sub}}$ according to Eqn. (2).
    Update the parameters of $F$ by: $F \leftarrow F - \alpha \nabla_F \mathcal{L}_{\text{cluster}}$ according to Eqn. (3).
    Re-assign the cluster label $\mathbf{y}_n$ to $\xi_n^{\text{sub}}$ based on the updated $F$.
    Update the cluster centers $\mathbf{C}$ according to Eqn. (4).
end
for $\xi \in \Xi$ do
    Take a uniformly sampled sub-trajectory $\xi^{\text{sub}}$ with length $l$ from $\xi$.
    Assign a cluster label $\mathbf{y}$ to $\xi$ according to Eqn. (2).
end
Output: The cluster label $\mathbf{y}$ of each trajectory $\xi$.
```

B Details for Contrastive Clustering

We further discuss the considerations of the design of contrastive clustering algorithm. Firstly, for varied-length sequences, it is difficult to design a proper distance metric to ensure that trajectories from the same mode are close, because common distance metrics such as per-step L2 or cosine distance on states cannot be used. Thus, contrastive learning is a good choice for learning the distance metric in a latent space for clustering. Secondly, separating contrastive learning and clustering into two stages may not find the optimal hidden space for clustering, while co-optimizing them can make them benefit from each other.

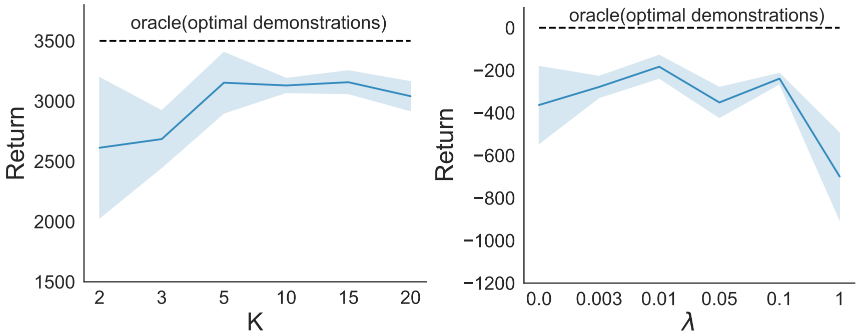
Implementation Details. For the implementation of the contrastive learning algorithm, in the subsampling step, we fix the length of the sub-trajectories, which is no longer than 50 steps since RNN usually suffers from catastrophic forgetting with long sequences. For the MuJoCo environment, the sub-trajectory length is fixed at 20. For the Driving environment, the sub-trajectory length is fixed at 15. For the Simulated Robot environment, the sub-trajectory length is fixed at 20. For training,

6th Conference on Robot Learning (CoRL 2022), Auckland, New Zealand.

we randomly sample the sub-trajectories with a fixed stride and make sure each sub-trajectory has the same length. After convergence, we use the representation of a sub-trajectory for clustering. We set the batch size of contrastive clustering as 128. We first pre-train the feature extractor only with the contrastive learning loss $\mathcal{L}_{\text{contrast}}$ for 200 iterations before initializing c_k and then train with the whole loss $\mathcal{L}_{\text{cluster}}$ for 2000 iterations. For the number of clusters K, we set it as 5, 10, and 10 at initialization for MuJoCo, Driving, and Robot Arm respectively. For the feature extractor, we use a one-layer LSTM model to extract representation for trajectories and set the dimension for the hidden state as 128. The hyper-parameter λ is fixed to 0.01 for all three environments. Learning rate α is fixed to 0.01 with Adam optimizer.

C Additional Experimental Results

C.1 Parameter Sensitivity



(a) Expected return w.r.t. cluster number K (b) Expected return w.r.t. trade-off weight λ

Figure 1: Sensitivity experiment results for hyperparameters K and λ

There are two key hyperparameters: cluster number K and the trade-off weight λ between L_{contrast} and L_{cluster} in our method. We investigate the sensitivity of the performance of our method to the hyperparameters. We show the expected return after convergence under different hyperparameters in Figure 1. The solid lines with shades show the mean and standard deviation of the expected return of our method and the dashed lines on top show the oracle optimal performance that the policy may achieve by only selecting and learning from the optimal demonstrations.

Results. We observe that when K is small, the converged model suffers from high variance and lower mean return, as a small number of clusters are not sufficient to capture all single modalities. Meanwhile, our method with larger K achieves consistent performance, because more clusters guarantee a clear separation between different modalities. Once the cluster number is enough to capture all the modalities, more clusters do not improve the performance. Nevertheless, we note that larger K brings extra computational cost since every cluster requires training a GAIL model, so we set K to 5, 10, and 10 respectively for 3 environments in consideration of the trade-off between efficiency and effectiveness. For the sensitivity of λ, we find that our framework works well under the value of λ ranging from 0.003 to 0.1, and $\lambda > 1$ leads to a severe drop in performance, mainly because too much emphasis on L_{cluster} will cause all samples to collapse into one or two clusters and the contrastive clustering algorithm becomes unable to separate different modalities.

Discussion on the choice of K. While in real scenarios when K is unknown to us, we can estimate it empirically by dimension reduction and then visualizing trajectories. If one wants to get an optimal K, a grid search around this approximation may be needed, but often an approximation is good enough. Note that the number of modes has no direct relationship with the number of source domains, especially in a real-world scenario: data can be collected every day, and each day can be seen as a source, but they may all fall into a certain number of modes, i.e., the number of modes will not increase unlimitedly. After contrastive clustering, transferability learning on each cluster can be done in parallel, which can save time. Only a subset of demonstrations can also be easier to fit, compared to fitting the whole dataset, which also boosts learning efficiency.

C.2 Visualization of Transferability

We visualize the transferability computed by the proposed method as well as all our baselines on the Driving environment. As is shown in Figure 2, the deeper the color, the higher transferability of the trajectory. We can observe that our method can mostly filter out non-transferable demonstrations (red arrow) for the target environment while assigning high transferability for transferable ones (green arrow).

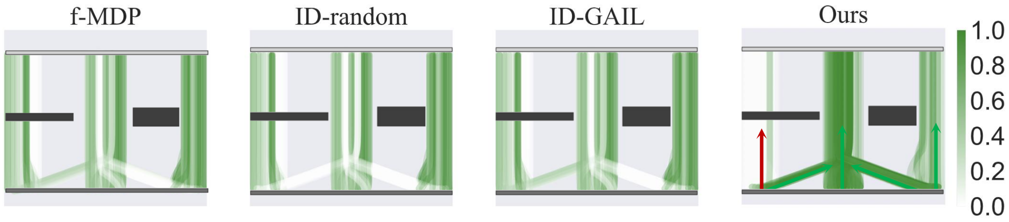


Figure 2: Visualization of the transferability in Driving for different methods.

C.3 Effect of the Ratio of Transferable and Non-transferable Demonstrations

To investigate the influence of the composition of the demonstrations on the final imitation performance, we conduct experiments on three MuJoCo environments with different ratios of the demonstrations from the four source demonstrators. For Hopper, we set the gravitational constant as (i) 15.0, (ii) 9.8, (iii) 2.0, (iv) 1.0. We fix the number of trajectories for (i) and (ii), i.e. relatively transferable demonstrations, and change the number of trajectories for (iii) and (iv). For Walker2d, we set friction to (i) 24.8, (ii) 9.9, (iii) 3.9, (iv) 1.1. For HalfCheetah, the compositions of demonstrations are set the same as in original paper, which is (i) $(1, 0.9)$, (ii) $(0.9, 1)$, (iii) $(1, 0.05)$, (iv) $(0.05, 1)$ with setting $(\cdot, \cdot)$ as the discount factor of the force of the front leg and the back leg. The ratio configurations and the results are shown in Table 1, Table 2, Table 3 respectively for three environments.

We observe that in the Hopper environment, with the increase of non-transferable trajectories, the performance of naive GAIL deteriorates and other baselines also drop dramatically due to the multimodal distribution effect, while our method shows stable performance with a high return against the changes in the composition of the source demonstrations. Moreover, comparing with the converged result of our method under different ratios, we observe that increasing non-transferable trajectories does not influence the final return of our method much, which indicates that our transferability measurement stably and accurately filters out non-transferable trajectories. Even for the easiest setting: $1:1:1:1$ with an equal number of transferable and non-transferable demonstrations, our method still outperforms GAIL. The results show that non-transferable demonstrations consistently influence imitation learning performance and measurement to filter out non-transferable demonstrations is important. Experiments on the other environments show similar results, demonstrating the robustness of our method under various scenarios.

Table 1: Results under different compositions of demonstrations in Hopper environment.

Composition	Naive GAIL	fMDP	ID w/o GAIL	ID w/ GAIL	Ours
$1:1:1:1$	2926±468	2947±412	1547±362	2287±315	**3259**±198
$1:1:2:2$	2845±360	2662±699	1335±787	2022±253	**3261**±206
$1:1:5:5$	2761±358	2361±537	1176±154	1042±730	**3104**±340
$1:1:10:10$	2137±685	2791±468	836±218	1314±412	**3049**±331
$1:1:25:25$	1083±244	1040±760	908±191	714±82	**3113**±413
$1:1:50:50$	739±184	1276±458	764±260	671±126	**2890**±556

C.4 Additional Experiments on Simulated Robot

We also conduct additional experiments on simulated Franka Panda Arm to better verify our proposed method. We create three demonstrators by disabling the No. $1, 3$ joints, the No. 1 joint, and using

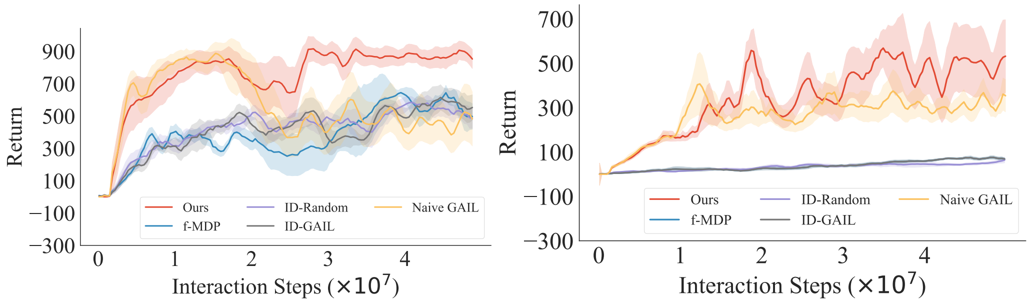
Table 2: Results under different composition of demonstrations in Walker2d environment.

Composition	Naive GAIL	fMDP	ID w/o GAIL	ID w/ GAIL	Ours
1 : 2 : 2 : 2	318±290	283±190	1688±218	1703±175	**2077**±216
1 : 2 : 5 : 5	288±72	249±37	345±92	328±39	**1731**±73
1 : 2 : 10 : 10	327±65	213±48	311±29	349±30	**1664**±166
1 : 2 : 20 : 20	287±74	339±131	345±73	320±64	**1629**±87

Table 3: Results under different composition of demonstrations in HalfCheetah environment.

Composition	Naive GAIL	fMDP	ID w/o GAIL	ID w/ GAIL	Ours
2 : 2 : 1 : 1	2389±897	404±246	2031±312	2210±86	**3008**±117
2 : 2 : 2 : 2	2882±84	247±308	2126±110	2067±86	**2997**±209
2 : 2 : 5 : 5	2201±502	1613±409	-327±119	1273±546	**3246**±134
2 : 2 : 10 : 10	2367±897	389±232	1808±146	1315±414	**2981**±71

fully-able joints respectively while disabling the No. $1, 3$ joints for the target imitator. We import demonstrations with the number of interaction steps 1×10^5, 1×10^5, and 1×10^5 for each source environment respectively. The reward function and the task are set as the same as that of the original task in the main paper. The result is shown in Fig. 3(a). Another setting is created similarly by disabling the No. $1, 3, 4, 6$ joints, the No. $1, 3$, and the No. 4 joint respectively while disabling the No. $1, 3, 4$ joints for the target imitator, with the result presented in Fig. 3(b).



(a)

(b)

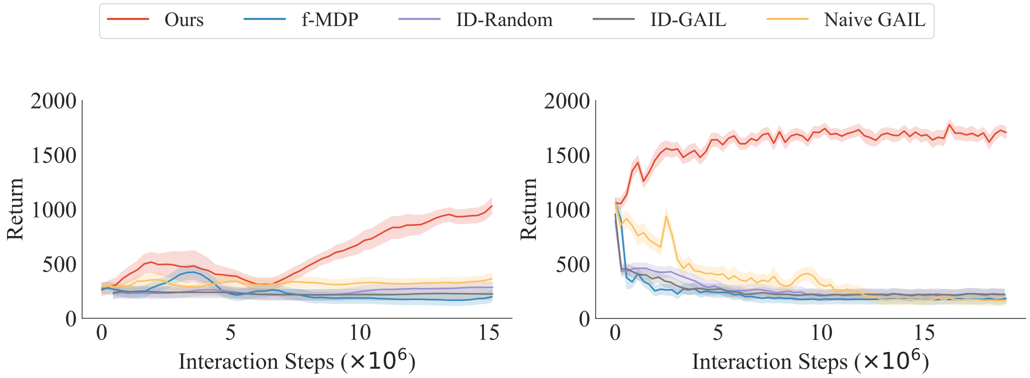
Figure 3: Additional experiment results on simulated robot environment.

C.5 Generalization to More Demonstrations

In real-world applications, there are situations where demonstrations in the original database are insufficient and new demonstrations are continuously collected from different sources to augment the database. We further demonstrate that the proposed method can use augmented demonstrations more effectively. We conduct experiments in the MuJoCo Walker2d experiment. We firstly collect 2, 2, 50 and 50 demonstrations from environment (i) 24.8, (ii) 9.9, (iii) 3.9, (iv) 1.1 respectively. The demonstrations are not enough to learn an optimal policy, but our method can still learn a transferability model and f-MDP and ID can learn a feasibility model. Then we add 10, 50, and 50 demonstrations from environment (v) 24.9, (vi) 0.7, and (vii) 0.1 respectively. Then we require all the methods not to re-train the transferability or the feasibility model but directly predict the transferability or feasibility for new demonstrations. The experiments aim to test the generalization ability of the model to filter out non-transferable demonstrations.

As shown in Fig. 4(a), when we only have insufficient demonstrations, we observe that the proposed method still achieves the highest point compared with other baselines, which demonstrates that we are able to use the demonstrations more efficiently even when they are insufficient.

Moreover, in Fig. 4(b), we are given new demonstrations. To use them selectively, we first use our contrastive-clustering LSTM model to assign a cluster label to each demonstration according to Eqn. (2). We then generate the transferability for the new demonstrations with the GAIL model in that cluster according to Eqn. (6). Note that we do not re-train the clustering model here with the new demonstrations but directly apply the clustering model and the GAIL model for transferability to cluster new demonstrations. For a fair comparison, we finetune the policy starting from the same checkpoint achieved by our method. The proposed method achieves the highest performance, which means that the proposed method possesses the capability of generalizing to unseen demonstrations. This generalization to new demonstrations can be extremely meaningful, which serves as a practical method to satisfy our intention of **continually** collecting more useful information from multiple sources. We do not require any extra computation other than a one-time inference, which is efficient to use.



(a) Results on insufficient trajectories.

(b) Generalization to newly given demonstrations.

Figure 4: Experiments for generalization to more demonstrations.

C.6 Comparison with a K-means Variant

To demonstrate the significance of our Sequence-based Contrastive Clustering algorithm, we conducted the following experiments on the Driving environment by K-Means clustering with the number of clusters K=10 (as the same in our method).

Specifically, we down-sampled each trajectory with a fixed stride to uniformly generate a fixed-length subsample, and applied the K-means algorithm directly to these sub-trajectories and therefore assign each trajectory to a cluster. Then, on each of these K clusters, we learn the transferability respectively. The result of using transferability generated by K-Means clustering for the final imitation learning is presented in Fig. 5.

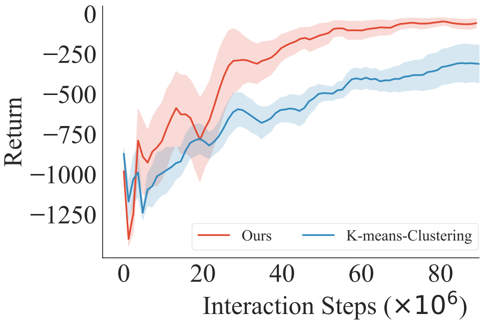


Figure 5: The ablation study on a K-means variant of our method.

We observed that the lacking of a contrastive learning step may cause difficulty in obtaining a high-quality unimodal clustering, which is essential for learning an accurate transferability measurement, and further cause a final performance drop. One way our contrastive clustering method is superior to K-means is that performing K-means on uniformly random-sampled sub-trajectories may introduce high variance into the clustering results, while our method, which makes different subsamples of the same trajectory as positive pairs and minimize their distance in the hidden representation space, can mitigate such instability. Also, the extracted representations are used for clustering, so it is beneficial if they are learned with the clustering step in a coherent manner.