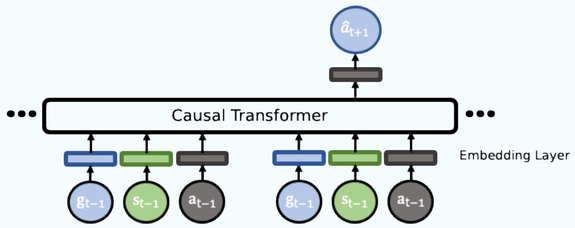
Contrastive Decision Transformers

Sachin Konan, Esmaeil Seraj, and Matthew Gombolay
CORE Robotics Lab, Institute for Robotics & Intelligent Machines (IRIM), Georgia Institute of Technology, Atlanta, GA, USA

CoRL 2022

Introduction: Decision Transformers

- Chen *et al*. (2021) introduced the Decision Transformer (DT): a return-conditioned transformer architecture for RL

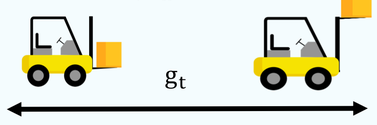


- DT abstracts the RL as a sequence modeling problem.
- Decision Transformer (DT):
  - **Inputs**: return (g_t), state (s_t), and action (a_t) tokens
  - **Outputs**: the right shifted prediction of the input, where ($\hat{a}_{t+1}$) is the action used in inference

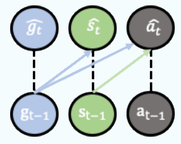
Motivation: Connecting Dots from Multi-Task Learning & RCRL

- Offline RL learns: $s_t \rightarrow a_t$, that maximizes g_{t+1}
- DT learns: $(g_{t-k}, s_{t-k}, a_{t-k}, \dots)$, g_t, $s_t \rightarrow a_t$

The distribution of g_t represents potentially *very different tasks*



g_t is **critical** because it is *encoded* into s_t and a_t

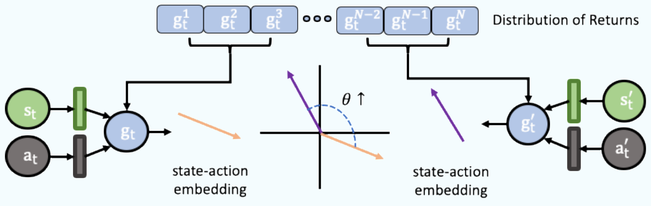


- From multi-task learning: Representations relating to different tasks should be distant
- For two sets of tokens (g_t, s_t, a_t) and (g'_t, s'_t, a'_t):

$\epsilon_{(g|s,a)}$ = *average distance of return/state-action embeddings*

$|g_t - g'_t| \leq \epsilon_g \rightarrow |s, a_t - s', a'_t| \leq \epsilon_{s,a}$

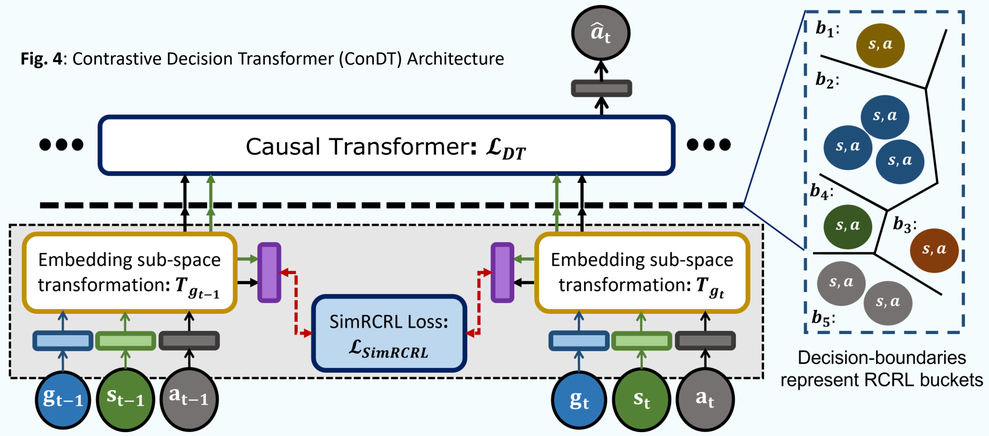
$|g_t - g'_t| \geq \epsilon_g \rightarrow |s, a_t - s', a'_t| \geq \epsilon_{s,a}$



- Cheung *et al*. (2019): distancing embeddings corresponding to different tasks can improve multi-task learning
- **Our Idea:** To help DT discriminate between different tasks, we want to use Return-Based Contrastive Learning (RCRL)
  - **RCRL**: maximize distance (θ) between state-action embeddings belonging to different g_t buckets

Contrastive Decision Transformers (ConDT): Model Architecture & Algorithmic Overview

Fig. 4: Contrastive Decision Transformer (ConDT) Architecture



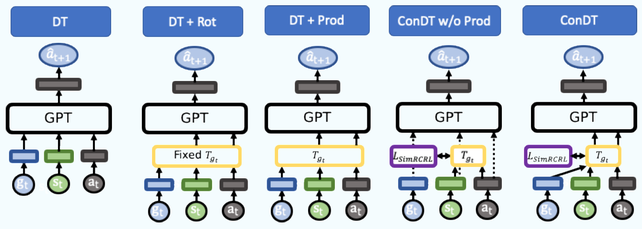
- We introduce **Contrastive Decision Transformers (ConDT)**, which builds on DT:
  1. We add a return-dependent transformation layer, T_{g_t}, that projects state and action embeddings
  2. We train T_{g_t}, using our new ConDT loss function ($L_{SimRCRL}$):

$$L_{SimRCRL} = \sum_{i=0}^{B_C} -\log\left(\frac{\exp\left(\left(z^i_{ah} \cdot z^i_p\right)/tau\right)}{\sum_j^B \mathbf{1}(i,j)\left[\exp\left(\frac{\left(z^i_{ah} \cdot z^j_{ah}\right)}{tau}\right) + \exp\left(\frac{\left(z^i_{ah} \cdot z^j_p\right)}{tau}\right)\right]}\right)$$

- We first sample B_C pairs of embeddings from different g_t buckets
- Here z_{ah}, z_p refer to the state-action embeddings of an anchor and its corresponding pair
- **Unlike RCRL**, $L_{SimRCRL}$ is a direct optimization of the distance between embeddings
- ConDT is trained with: $L_{DT} + \beta * L_{SimRCRL}$, where L_{DT} is the general DT loss and β weighs $L_{SimRCRL}$

Testing & Experiments: Training Ablations & Evaluation Environments

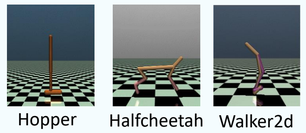
- To test the effectiveness of T_{g_t} and $L_{SimRCRL}$ we evaluated 5 baselines:



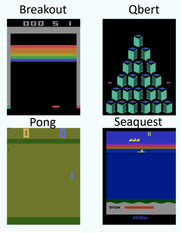
Summary of Differences:
1. *DT* – Baseline DT
2. *DT+Rot* – T_{g_t} is a random rotation using g_t
3. *DT+Prod* - T_{g_t} is a learnable transformation
4. *ConDT w/o Prod* - T_{g_t} is used to train embeddings
5. *ConDT* - T_{g_t} is a learnable transformation, trained with $L_{SimRCRL}$

- We evaluated ConDT across three domains:
  - I. Open AI Gym,
  - II. Atari 2600,
  - III. Adroit Robotic Handgrip

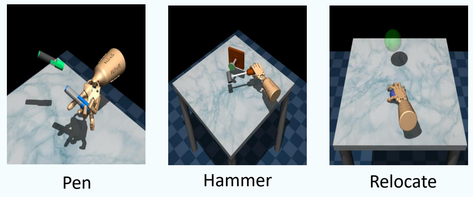
Open AI Gym



Atari 2600

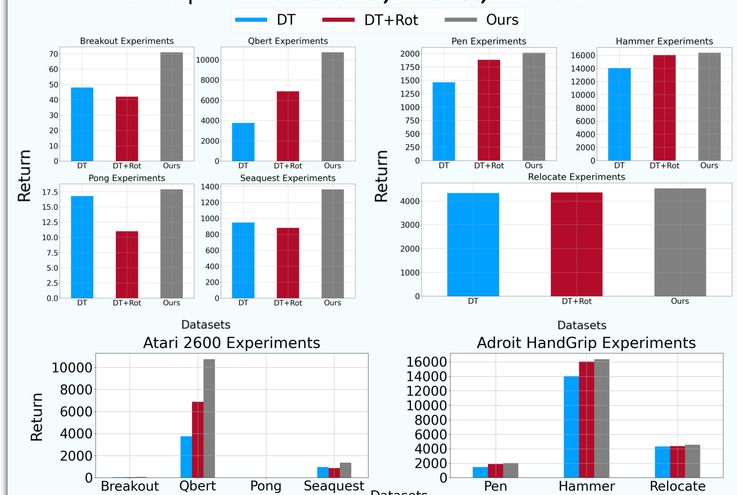


24-DoF Adroit Robotic Handgrip

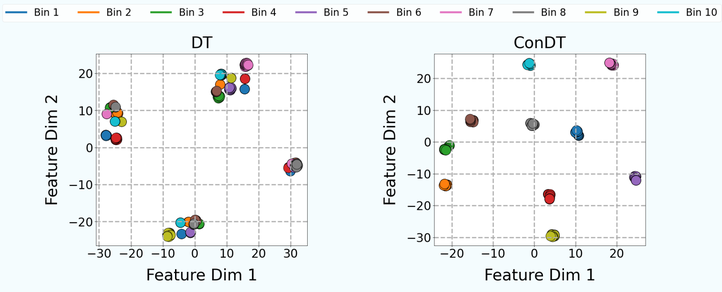


Empirical Evaluation: Results & Experiments

- We show the performance of **DT, DT+Rot,** and **ConDT:**



- **In summary**, across all experiments, ConDT results in better performance in all domains. ConDT even results in 3× return gain in the Qbert Atari Experiments. Also, DT+Rot confirms that distancing representations can achieve sizable return.
- **Ablation Study**: We investigate how well ConDT distances its representations relative to DT:



- **In summary**, ConDT embeddings are not only clustered with respect to their own return bin, but they are also spread w.r.t other bins (i.e., higher positive and smaller negative similarity).

Conclusions

- DT (Chen *et al*. 2021) experimentation showed promise, while there still lies a performance gap between DT and SOTA offline RL methods.
- We proposed **Contrastive Decision Transformers (ConDT)**. ConDT adds a return-based transformation layer, trained with $L_{SimRCRL}$.
- ConDT beats DT in several experiments across **OpenAI Gym, Atari, and Adroit Robotic Handgrip Manipulation** domains.
- $L_{SimRCRL}$ experimentally distances state-action embeddings by return.

- Demo:
- Code:
- Full-read: