Numerical Goal-based Transformers for Practical Conditions

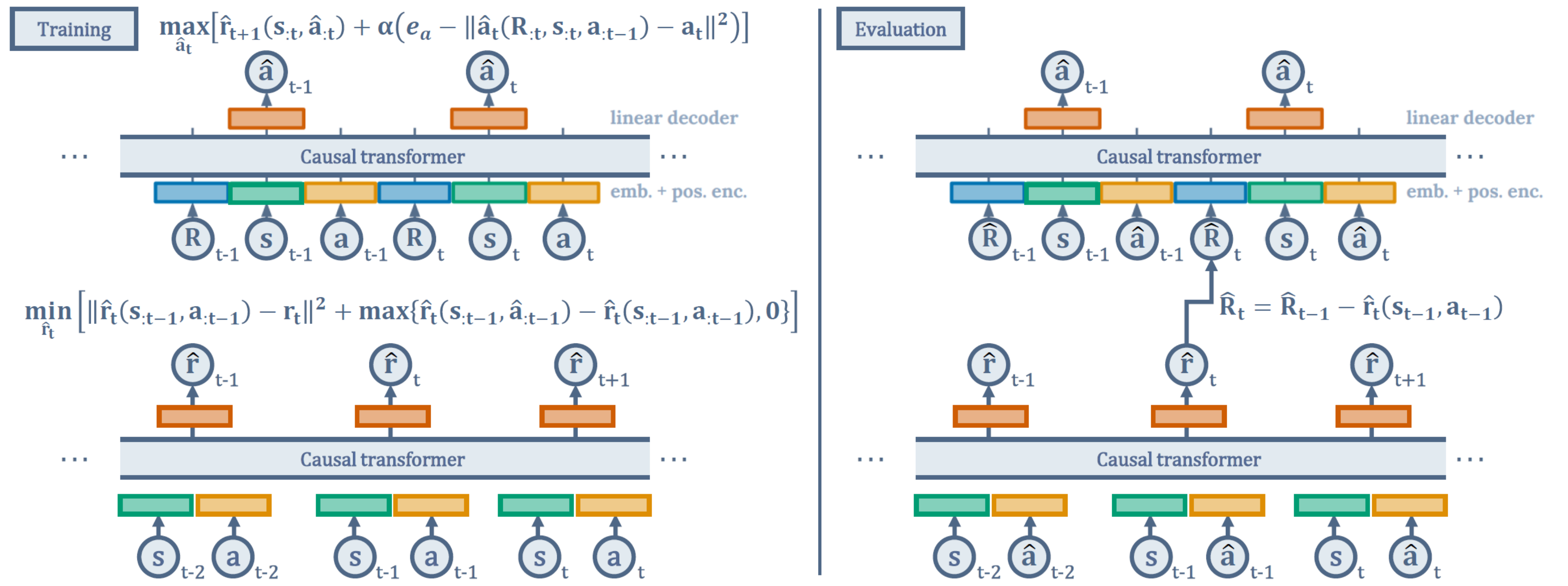
Seonghyun Kim, Samyeul Noh, Ingook Jang*
{kim-sh, samuel, ingook}@etri.re.kr

1. Motivation

Goal-conditioned reinforcement learning (GCRL) studies aim to apply trained agents in **realistic environments**. In particular, offline reinforcement learning is being studied as a way to reduce the cost of online interactions in GCRL.
One such method is **Decision Transformer (DT)**, which utilizes a **numerical goal called "return-to-go"** for superior performance. Since DT assumes **an idealized environment**, such as **perfect knowledge of rewards**, it is necessary to study an improved approach for real-world applications.
In this work, we present various attempts and results for numerical goal-based transformers to operate under practical conditions.

2. Conservative Decision Transformer

A. The architecture of Conservative Decision Transformer


Training
$\max_{\hat{a}_t}[\hat{r}_{t+1}(s_{:t}, \hat{a}_{:t}) + \alpha(e_a - \|\hat{a}_t(R_{:t}, s_{:t}, a_{:t-1}) - a_t\|^2)]$
$\hat{a}_{t-1}$
$\hat{a}_t$
linear decoder
Causal transformer
emb. + pos. enc.
R_{t-1} s_{t-1} a_{t-1} R_t s_t a_t
$\min_{\hat{r}_t}[\|\hat{r}_t(s_{:t-1}, a_{:t-1}) - r_t\|^2 + \max\{\hat{r}_t(s_{:t-1}, \hat{a}_{:t-1}) - \hat{r}_t(s_{:t-1}, a_{:t-1}), 0\}]$
$\hat{r}_{t-1}$ $\hat{r}_t$ $\hat{r}_{t+1}$
Causal transformer
s_{t-2} a_{t-2} s_{t-1} a_{t-1} s_t a_t
Evaluation
$\hat{a}_{t-1}$
$\hat{a}_t$
linear decoder
Causal transformer
emb. + pos. enc.
$\hat{R}_{t-1}$ s_{t-1} $\hat{a}_{t-1}$ $\hat{R}_t$ s_t $\hat{a}_t$
$\hat{R}_t = \hat{R}_{t-1} - \hat{r}_t(s_{t-1}, a_{t-1})$
$\hat{r}_{t-1}$ $\hat{r}_t$ $\hat{r}_{t+1}$
Causal transformer
s_{t-2} $\hat{a}_{t-2}$ s_{t-1} $\hat{a}_{t-1}$ s_t $\hat{a}_t$

In the training phase, **all data** from the offline dataset, including returns and rewards, are utilized to update the networks. **In the evaluation phase**, considering practical conditions, **only states** are used to interact with the environment.

B. Objective functions for Conservative Decision Transformer

1) The objective function for the reward estimation

$$J_\theta = \min_\theta \Big\{ \|\widehat{r}_t(s_{:t-1}, a_{:t-1}) - r_t\|^2 + \max\big(\widehat{r}_t(s_{:t-1}, \widehat{a}_{:t-1}) - \widehat{r}_t(s_{:t-1}, a_{:t-1}), 0\big) \Big\}, \qquad (1)$$

Since the objective function in Eq. (1) is a minimization problem, the optimal value of the second term in Eq. (1) is 0. This means that the **estimated reward for unseen actions** should be **conservatively lower** than the reward for actions in the offline dataset, i.e., $\widehat{r}_t(s_{:t-1}, \widehat{a}_{:t-1}) \le \widehat{r}_t(s_{:t-1}, a_{:t-1})$.

2) The objective function for the action generation

$$J_\phi = \max_\phi \Big\{ \widehat{r}_{t+1}\big(s_{:t}, \widehat{a}_{:t}(R_{:t}, s_{:t}, a_{:t-1})\big) + \alpha\big(e_a - \|\widehat{a}_t(R_{:t}, s_{:t}, a_{:t-1}) - a_t\|^2\big) \Big\}, \qquad (2)$$

The objective function in Eq. (2) means that **a generated action** should be found by considering **maximizing the estimated reward** and **minimizing the action error** within error threshold e_a. When α is large, the objective function tends to minimize the action error, and when α is close to 0, it tends to maximize the estimated reward.

3) Objective function for alpha

The α is automatically adjusted by solving a **Lagrangian dual problem of Eq. (2):**

$$J_\alpha = \min_\alpha \alpha\big(e_a - \|\widehat{a}_t(R_{:t}, s_{:t}, a_{:t-1}) - a_t\|^2\big). \qquad (4)$$

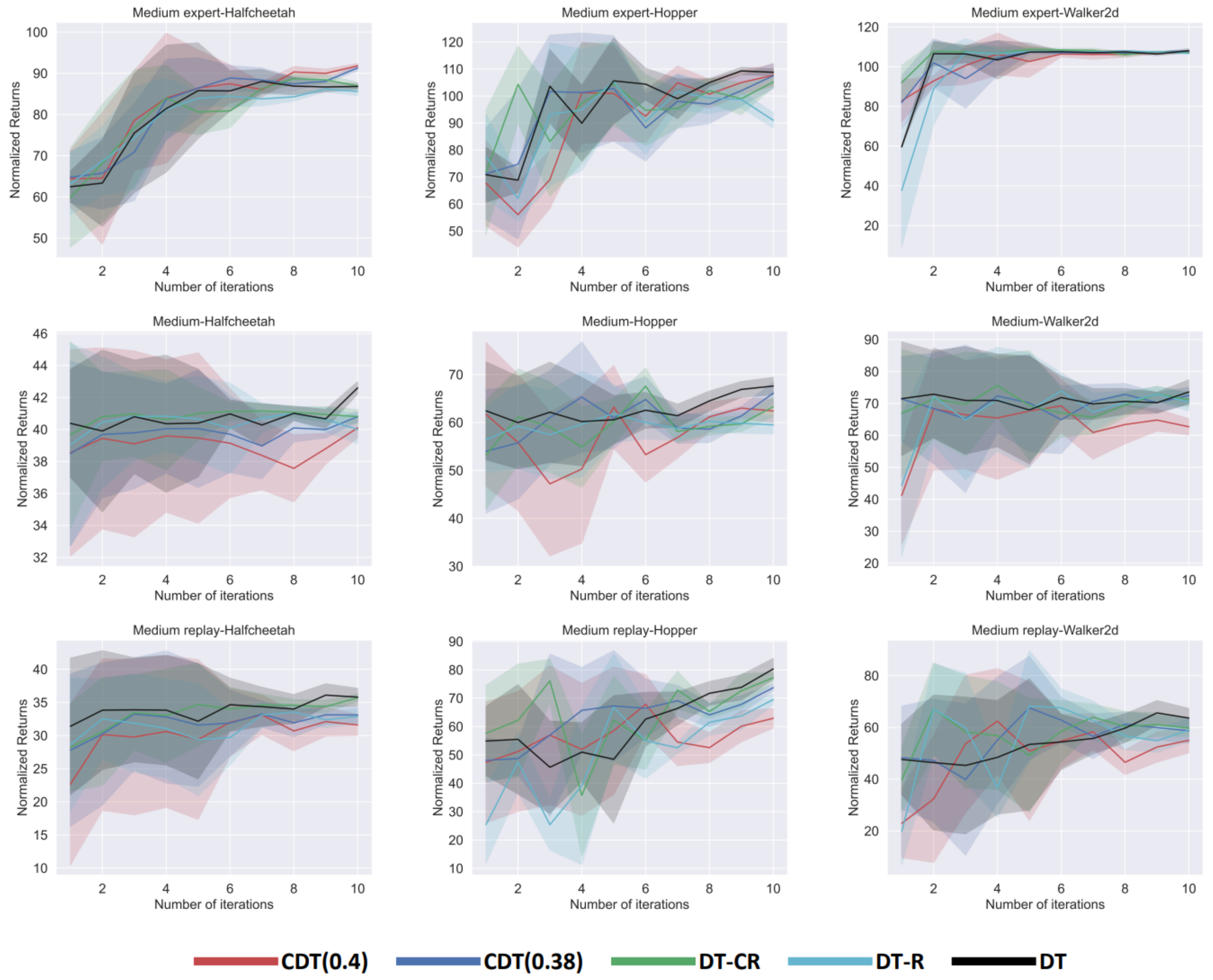
Since the objective function in Eq. (4) is a minimization problem, α is going to 0 for the condition $\|\widehat{a}_t(R_{:t}, s_{:t}, a_{:t-1}) - a_t\|^2 \le e_a$. For the opposite condition, α is going to be larger.

C. The algorithm for Conservative Decision Transformer

```
Algorithm 1 Conservative Decision Transformer
Initialize parameters $\theta$, $\phi$ and $\alpha$.
Load offline dataset $\mathcal{D} = \{(R_t, s_t, a_t, r_{t+1}) | (R_t, s_t, a_t, r_{t+1})$ over all $t$ in all episodes.$\}$
for each iteration do
    Sample $K$ trajectories with batch size $B$:
        $\mathcal{D}_{K,B} \sim \mathcal{D}(R_{t:t+K-1}, s_{t:t+K-1}, a_{t:t+K-1}, r_{t+1:t+K})$
    for each gradient step do
        Update the reward estimation:
            $\theta \leftarrow \theta - \lambda \nabla_\theta J_\theta$ using Eq. (1)
        Update the action generation:
            $\phi \leftarrow \phi - \lambda \nabla_\phi J_\phi$ using Eq. (2)
        Update the weight:
            $\alpha \leftarrow \alpha - \lambda_\alpha \nabla_\alpha J_\alpha$ using Eq. (4)
    end for
end for
```

3. Experimental Results

A. Figure: Training curves of all algorithms in Mujoco locomotion tasks



CR: Conservative Reward estimation, R: Reward estimation

1) Comparisons of CDT with ea = 0.4 and ea=0.38

In the figure, from the Medium Expert to the Medium Replay of datasets, **CDT with ea = 0.4 has higher variances** than those of **CDT with ea = 0.38**.
It means that this **high variance of CDT with ea = 0.4 due to relaxed error threshold** has **positive effects on good quality datasets**, such as the **Medium Expert dataset**.

B. Table: Normalized returns for all algorithms in Mujoco locomotion tasks

Dataset	Environment	CDT e_a =0.4	CDT e_a =0.38	DT-CR	DT-R	DT	CQL
Medium Expert	HalfCheetah	**91.8**	91.3	86.9	85.6	86.8	62.4
	Hopper	**107.7**	107.5	105.2	91.0	107.6	111.0
	Walker2d	107.2	**107.9**	107.0	106.7	108.1	98.7
Medium	HalfCheetah	40.1	**40.8**	**40.8**	40.0	42.6	44.4
	Hopper	62.4	**66.1**	63.3	59.5	67.6	58.0
	Walker2d	62.7	**72.5**	71.2	70.4	74.0	79.2
Medium Replay	HalfCheetah	31.6	33.1	**35.7**	32.9	36.6	46.2
	Hopper	62.9	73.7	**77.2**	69.5	82.7	48.6
	Walker2d	55.0	58.7	**59.7**	59.0	66.6	26.7

1) Comparisons of the proposed algorithms and DT

For most datasets and environments**, the proposed algorithms** have **similar performance with slightly lower values than those of DT due to estimation errors**.

2) Comparisons of CDT and DT

For the **Medium Expert dataset**, it is observed that CDT has better or comparable performance to DT. However, for the Medium and Medium Replay datasets, it is observed that the performance gap between CDT and DT gradually increases. These results show that, from the **perspective of the quality of the dataset**, the performance of the **proposed algorithm decreases** as the **proportion of highly rewarded trajectories in the overall dataset decreases** and the consistency of actions decreases.

3) Comparisons of DT-R and DT-CR

Over the entire datasets and environments, it is shown that the **reward estimation** to minimize the error **in a conservative manner outperforms** the reward estimation to minimize **the error only**.

4) Comparisons of CDT and DT-CR

For the **Medium Expert and Medium datasets**, it is observed that **CDT has higher performance** because **it generates actions by considering the maximization of the conservatively estimated reward.**
However, for the Medium Replay dataset, it is observed that **DT-CR outperforms CDT**. This means that if the dataset has few trajectories with high rewards and low consistency of actions in trajectories, **it is better to only minimize the action error** rather than maximize the conservatively estimated reward.

4. Conclusion

In this work, **we propose CDT for numerical goal-based transformers to operate in practical environments**. Experimental results show that the **CDT can achieve stable performance with only state information** and no actual reward information.
In future work, we would like to study a **generalized GCRL for 3D locomotion and robot manipulation tasks** by considering **various types of goals such as images, text, symbols, etc.**