澳門大學
UNIVERSIDADE DE MACAU
UNIVERSITY OF MACAU

Crossway Diffusion: Improving Diffusion-based Visuomotor Policy via Self-supervised Learning

GONG Ying
gong.ying@connect.umac.mo

2024.1.19

Contents

1. Introduction

- Behavioral cloning (BC) works well when a sufficient amount of training data is provided.
- Recently sequence modeling approaches have been often used for BC, of which the objective is to model the probability distribution of the multi-step state-action trajectory.
- For visuomotor control tasks, “Diffusion Policy” demonstrated promising performance using multimodal states including visual observations as the conditions of the diffusion model.
- In this work Crossway Diffusion is proposed. It’s a simple yet effective method to enhance diffusion-based visuomotor policy learning via a state decoder and a self-supervised learning (SSL) objective.

1. Introduction

Contributions:

- Crossway Diffusion is proposed, improving diffusion-based visuomotor policy via a state decoder and a simple SSL objective.
- The effectiveness of the method is confirmed on multiple challenging visual BC tasks from different benchmarks, including 2 real-world robot manipulation datasets.
- Detailed ablations are conducted on multiple design choices, verifying the advance and robustness of the proposed design.

2. Preliminaries

A. Behavioral Cloning

- Simple behavioral cloning (BC) sets over a Markov Decision Process (MDP), described by the tuple (S, A, P).
- The goal is to train a robot policy π that best recovers an unknown policy π^* using a demonstration dataset D = $\{(s_i, a_i)\}$ collected by π^*. Specifically, the robot policy π operates on a trajectory basis: $\pi(A_t|S_t)$, where $S_t = \{s_{t-T_s+1}, s_{t-T_s+2}, \dots, s_t\}$ is the given short history state sequence and $A_t = \{a_t, a_{t+1}, \dots, a_{t+T_a-1}\}$ is the predicted future actions to take.

2. Preliminaries

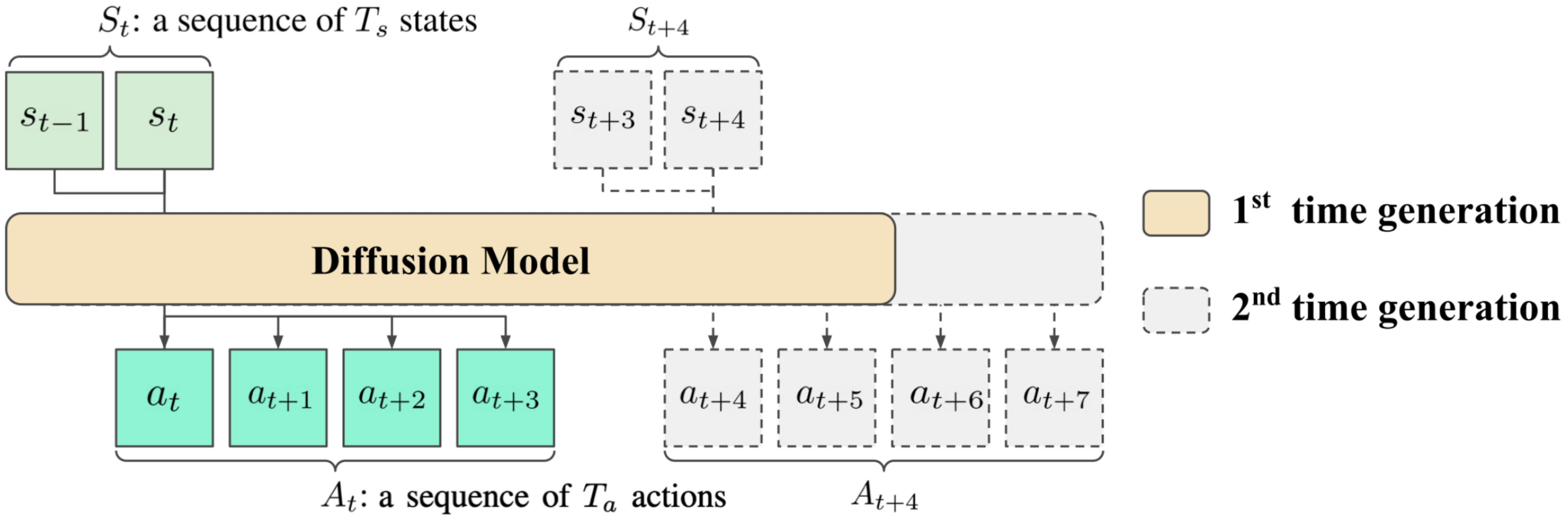
B. Diffusion Models

- Diffusion models are generative models that iteratively generate samples that match the data distribution.
- Forward process:
  The original data is destroyed by a sequence of noise $q(x^k|x^{k-1})$.
- Backward process:
  $p_\theta(x^{k-1}|x^k)$. is used to denoise the corrupted data.

2. Preliminaries

C. Diffusion Models for Policy Learning

- "Diffusion policy: Visuomotor policy learning via action diffusion" is feasible for visuomotor policy learning despite high dimensionality of the visual observations, by generating only action sequences, while conditioned on visual and other states.



$T_s = 2$ and $T_a = 4$

2. Preliminaries

C. Diffusion Models for Policy Learning

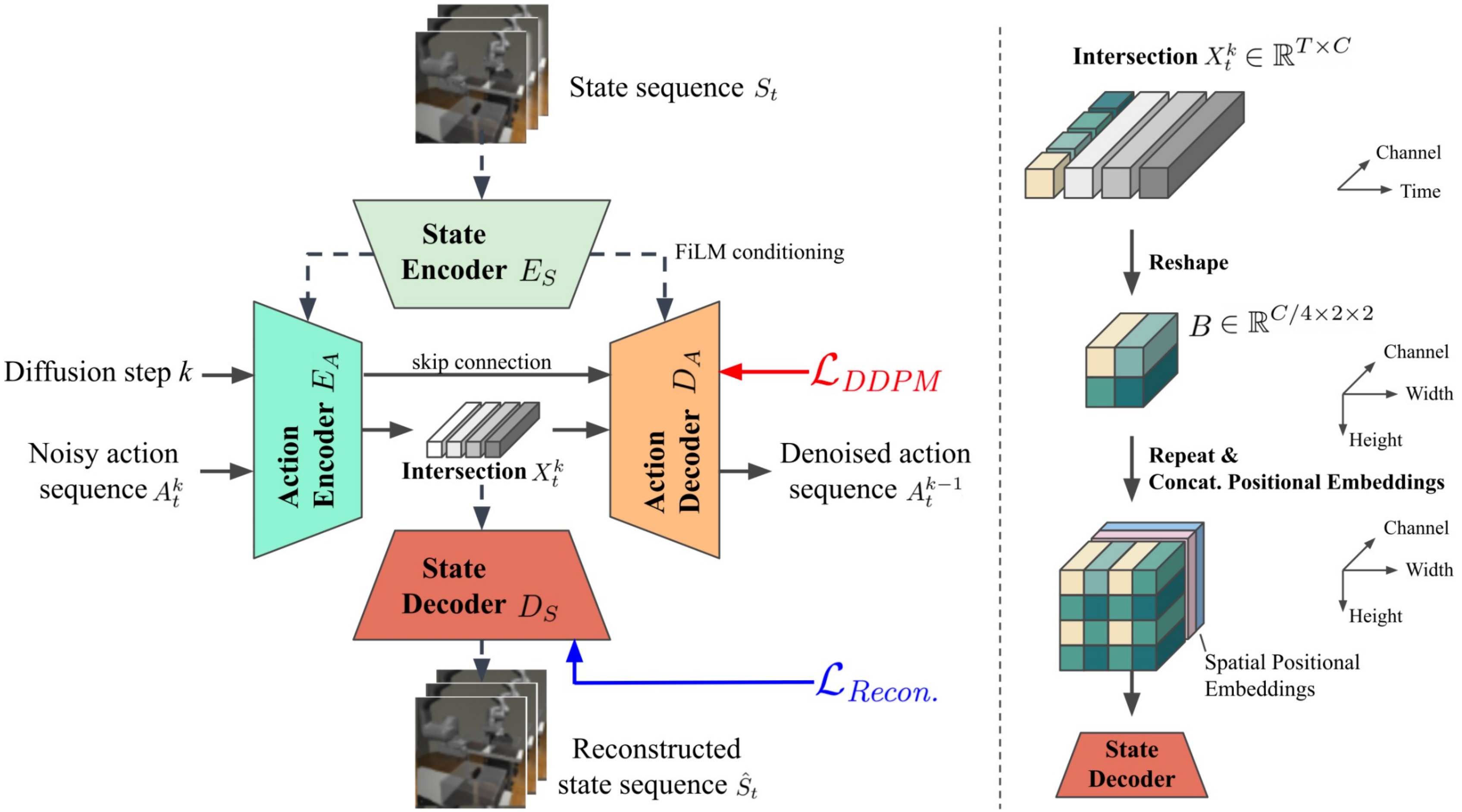
- Given a state sequence with both visual and low dimensional states $S_t = \{S_{t,img}, S_{t,low-dim}\}$, the state encoder extracts visual embeddings from images $h_{t,img} = E_S(S_{t,img})$. The visual embeddings $h_{t,img}$ are then concatenated with other low dimensional states $S_{t,low-dim}$ to form the observation condition $h_t = h_{t,img} \oplus S_{t,low-dim}$.

3. Method

Crossway Diffusion extends existing "Diffusion Policy" by (1) a state decoder and (2) an auxiliary objective, both for reconstructing input states.

The state decoder takes the intermediate representation of the diffusion process X_t^k to reconstruct the input states.

The reconstruction objective is jointly optimized with the diffusion loss L_{DDPM} during training.



3. Method

A. State Decoder

$$\hat{S}_t = D_S(g(X_t^k)),$$

where $g(\cdot)$ is the intersection transformation

a) Reconstruct the visual states
- The visual state encoders are are made of a sequence of 2D residual convolutional blocks, transposed convolutional layers for upsampling (ConvTranspose), and vanilla convolutional layers.
  - **Transposed conv layers**: upsample
  - **Vanilla conv layers**: extract features
- **Positional embedding**: convert pixel coordinates to learnable vector representations

b) Reconstruct the low-dimensional states
- Low-dim states are regressed by 3-layer multilayer perception (MLP) to help the model better learn low-dim representations, thus capture key information of input states.

3. Method

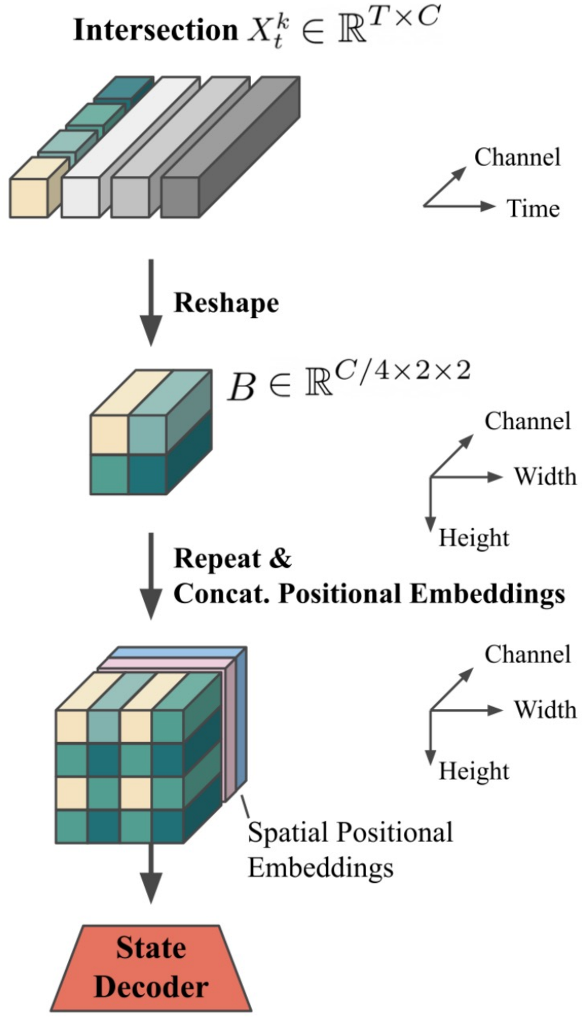
B. Intersection Transformation

$g(\cdot)$ converts the intermediate representation into original low-dim and visual states for reconstruction by state decoder.

a) $\boldsymbol{g_{low-dim}}(\cdot)$

Identity function.

b) $\boldsymbol{g_{img}}(\cdot)$



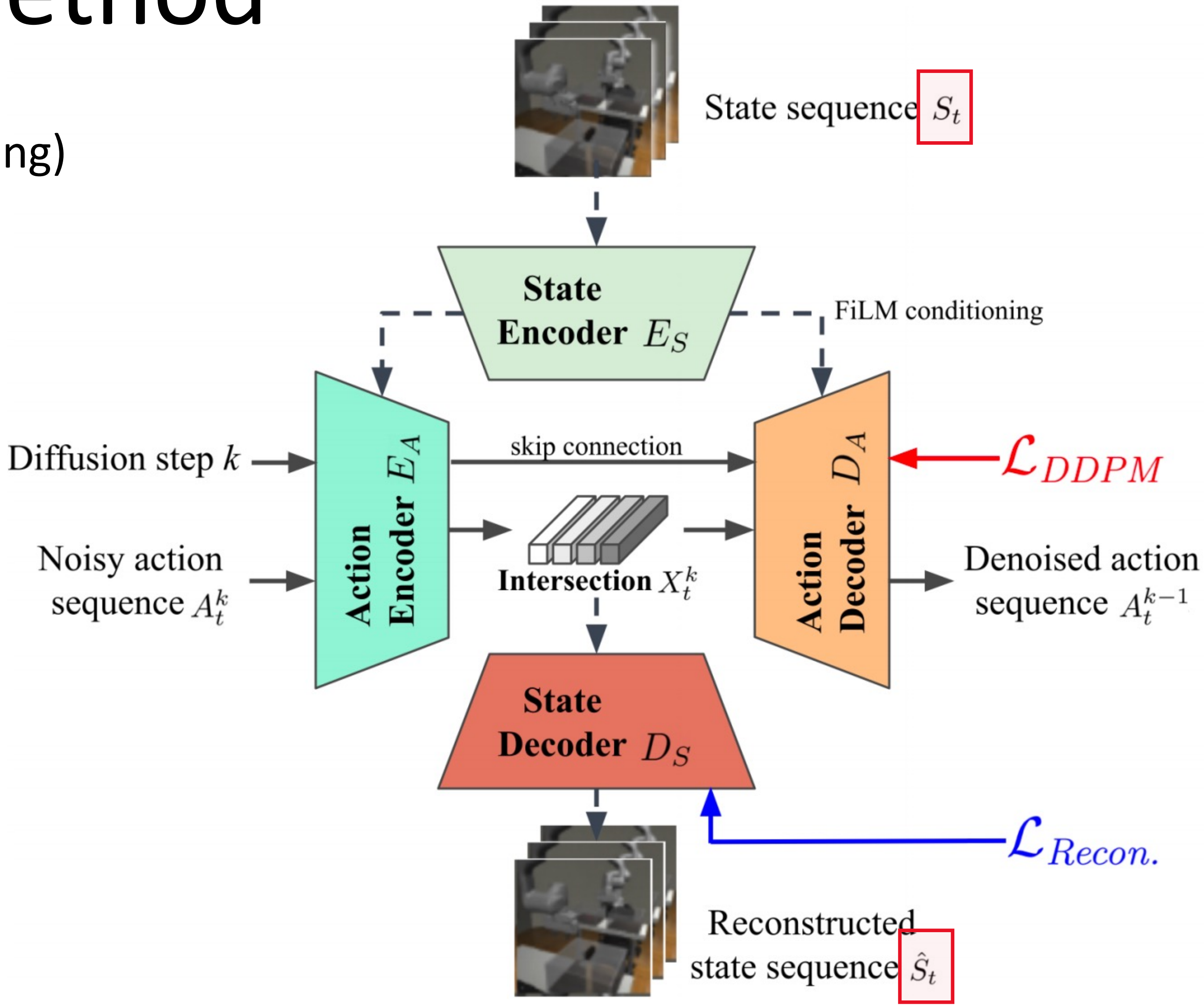
3. Method

C. Crossway Diffusion Loss (self-supervised learning)

$$L_{Recon.} = MSE(S_t, \hat{S}_t)$$

$$L_{Crossway} = L_{DDPM} + \alpha L_{Recon.}$$

- α = 0.1 is found to be a generally good setting without extensive hyperparameter search.



4. Experiment

A. Dataset summary

Task	ph	mh	R?	Rob.	Obj.	Cam.	Act-D	Steps
Can	200	300	N	1	1	2	7	400
Lift	200	300	N	1	1	2	7	400
Square	200	300	N	1	1	2	7	400
Transport	200	300	N	2	3	4	14	700
Tool Hang	200	-	N	1	2	2	7	700
Push-T	200	-	N	1	1	1	2	300
Duck Lift	100	-	Y	1	1	2	4	50
Duck Collect	100	-	Y	1	1	2	4	200

4. Experiment

B. Scores comparisons on simulated datasets

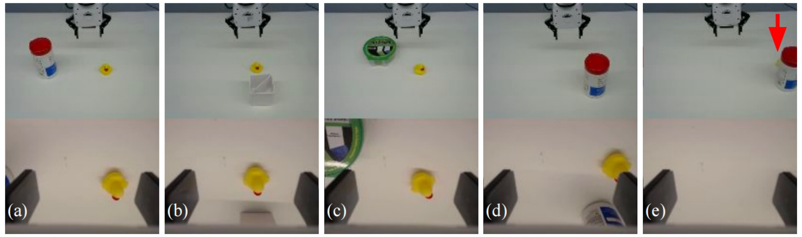
Method	Can, ph	Can, mh	Lift, ph	Lift, mh	Square, ph	Square, mh	Transport, ph	Transport, mh	Tool Hang, ph	Push-T
LSTM-GMM	0.714 ± 0.247	0.887 ± 0.033	0.978 ± 0.017	0.992 ± 0.001	0.643 ± 0.023	0.491 ± 0.057	0.656 ± 0.049	0.254 ± 0.017	0.460 ± 0.060	0.567 ± 0.013
IBC [32]	0.008 ± 0.006	0.001 ± 0.001	0.709 ± 0.008	0.222 ± 0.112	0.002 ± 0.001	0.000 ± 0.001	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.687 ± 0.031
Diffusion Policy CNN [17]	0.992 ± 0.002	0.958 ± 0.003	**1.000 ± 0.000**	**0.998 ± 0.001**	**0.935 ± 0.006**	0.858 ± 0.007	0.859 ± 0.015	0.643 ± 0.004	0.772 ± 0.012	0.819 ± 0.002
Crossway Diffusion (Ours)	**0.994 ± 0.002**	**0.965 ± 0.003**	**1.000 ± 0.000**	**0.998 ± 0.000**	**0.935 ± 0.005**	**0.879 ± 0.010**	**0.864 ± 0.016**	**0.800 ± 0.020**	**0.792 ± 0.014**	**0.843 ± 0.020**

The average of 3000 episodes and the standard deviation of 3 seeds.

C. Success rate of real-world tasks

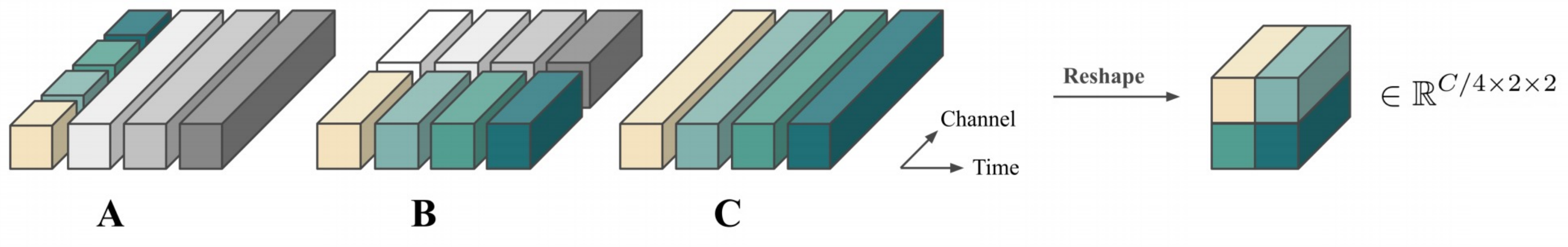
TABLE III: Success rate of real-world tasks

	Duck Lift	Duck Collect
Diffusion Policy CNN [17]	0.80	0.70
Crossway Diffusion (Ours)	**0.95**	**0.80**



4. Experiment

D. Ablations – on state decoder



Design D utilizes h_t instead of intersection X_t^k for reconstruction.

	Square, mh	Transport, ph	Transport, mh	Tool Hang, ph	Push-T
A (default)	0.879 ± 0.010	0.864 ± 0.016	0.800 ± 0.020	**0.792 ± 0.014**	**0.843 ± 0.020**
B	**0.881 ± 0.017**	0.882 ± 0.010	0.784 ± 0.025	0.777 ± 0.010	0.835 ± 0.012
C	0.868 ± 0.006	**0.906 ± 0.012**	**0.814 ± 0.028**	0.783 ± 0.005	0.831 ± 0.003
D	0.873 ± 0.012	0.892 ± 0.002	0.764 ± 0.013	0.790 ± 0.007	0.819 ± 0.015
Diff. [17]	0.858 ± 0.007	0.859 ± 0.015	0.643 ± 0.004	0.772 ± 0.012	0.819 ± 0.002

4. Experiment

D. Ablations – on auxiliary objective

Default	Shallower Dec.	ViT Dec.	Visual-only	Diff. [17]
0.843 ± 0.020	0.822 ± 0.014	0.824 ± 0.008	0.828 ± 0.012	0.819 ± 0.002

$N = 0$ (default)	$N = 2$	$N = 4$	$N = 6$	$N = 8$
0.843 ± 0.020	0.818 ± 0.006	0.827 ± 0.014	0.817 ± 0.003	0.803 ± 0.013

	Lift, mh	Lift, ph	Square, mh	Square, ph	Push-T
Crossway-CURL	0.802 ± 0.024	0.678 ± 0.188	0.053 ± 0.025	0.035 ± 0.007	0.518 ± 0.160
Default	**0.998 ± 0.000**	**1.000 ± 0.000**	**0.879 ± 0.010**	**0.935 ± 0.005**	**0.843 ± 0.020**
Diff. [17]	**0.998 ± 0.001**	**1.000 ± 0.000**	0.858 ± 0.007	0.935 ± 0.006	0.819 ± 0.002

Crossway-CURL adopts a contrastive loss as the auxiliary objective.

Thank you.