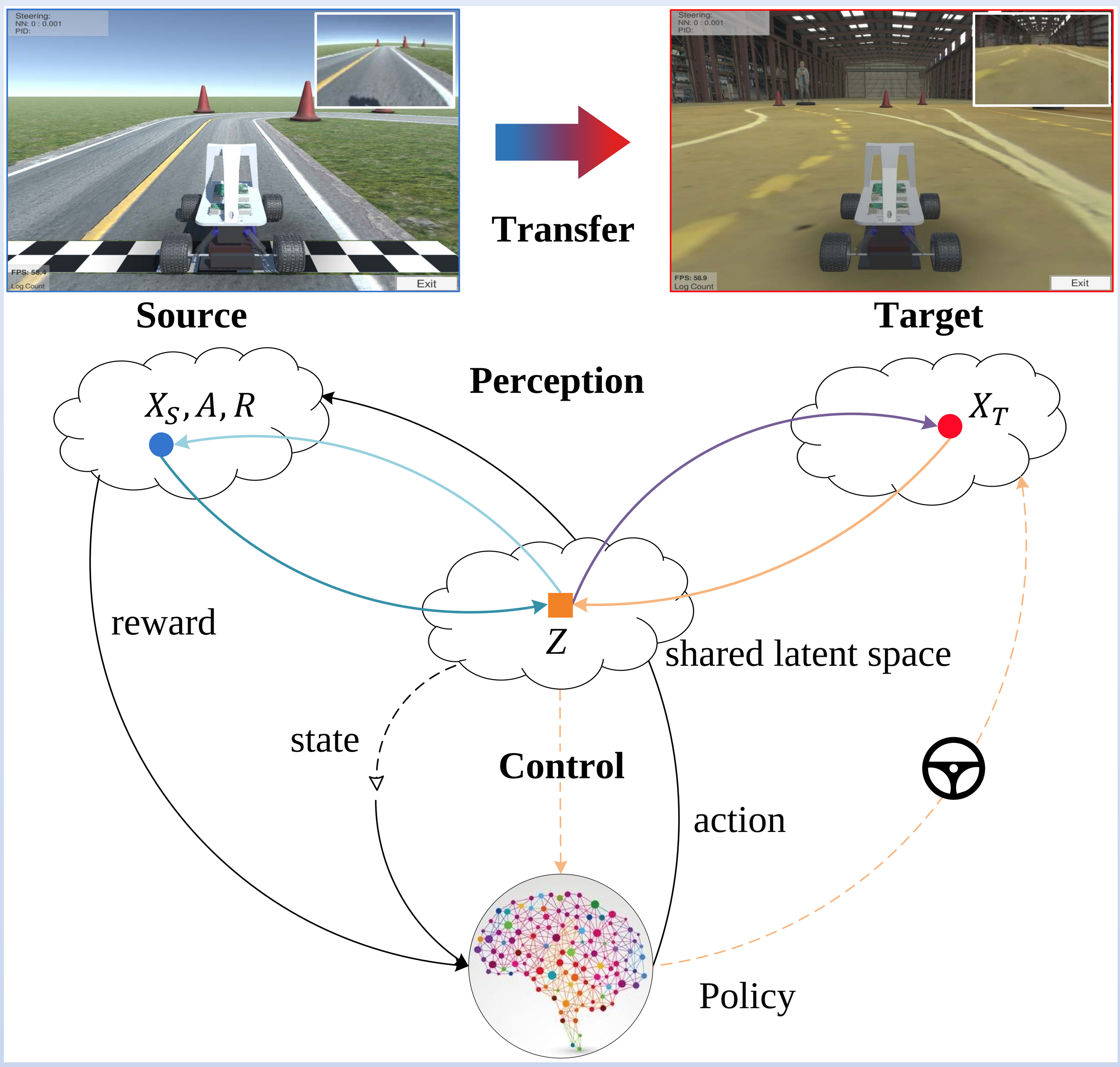
Deep Reinforcement Learning for Autonomous Driving by Transferring Visual Features

Hongli Zhou[1], Xiaolei Chen[2], Guanwen Zhang[1]*, Wei Zhou[1]

1 School of Electronics and Information, Northwestern Polytechnical University, Xi'an, China

2 CNPC logging Co.,Ltd, China



Introduction

We propose an effective transfer model that combines adversarial training (Perception) with reinforcement learning (Control).

a) Let X_S and X_T be samples from the source and target scenes for perceptual training (colored lines).

b) Define the reward (R) and the action (A) of the source environment for control training (black lines).

c) Perception and control systems are trained synchronously to jointly learns a driving policy from the common latent space Z.

d) This method does not require feedback from the target scene, yet learns a policy which can be applied directly to the target setting (yellow lines).

Methods

We divide the traditional DRL module into two subsystems: perception and control. The perceptual system maps visual images to a shared latent space by various constraints. The control system trains driving policy with reinforcement learning in the low-dimensional state.

a) Reconstruction loss: $\mathcal{L}_{re} = \mathbb{E}_{x_S\sim P_S}[\| x_S^{re} - x_S \|_1] + \mathbb{E}_{x_T\sim P_T}[\| x_T^{re} - x_T \|_1]$

b) Translation loss: $\mathcal{L}_{tr} = \mathbb{E}_{x_S\sim P_S}\left[(D_T(x_T^{tr}) - c_S)^2\right] + \mathbb{E}_{x_T\sim P_T}\left[(D_S(x_S^{tr}) - c_T)^2\right]$

c) Cycle consistency loss: $\mathcal{L}_{cy} = \mathbb{E}_{x_S\sim P_S}[\| x_S^{cy} - x_S \|_1] + \mathbb{E}_{x_T\sim P_T}[\| x_T^{cy} - x_T \|_1]$

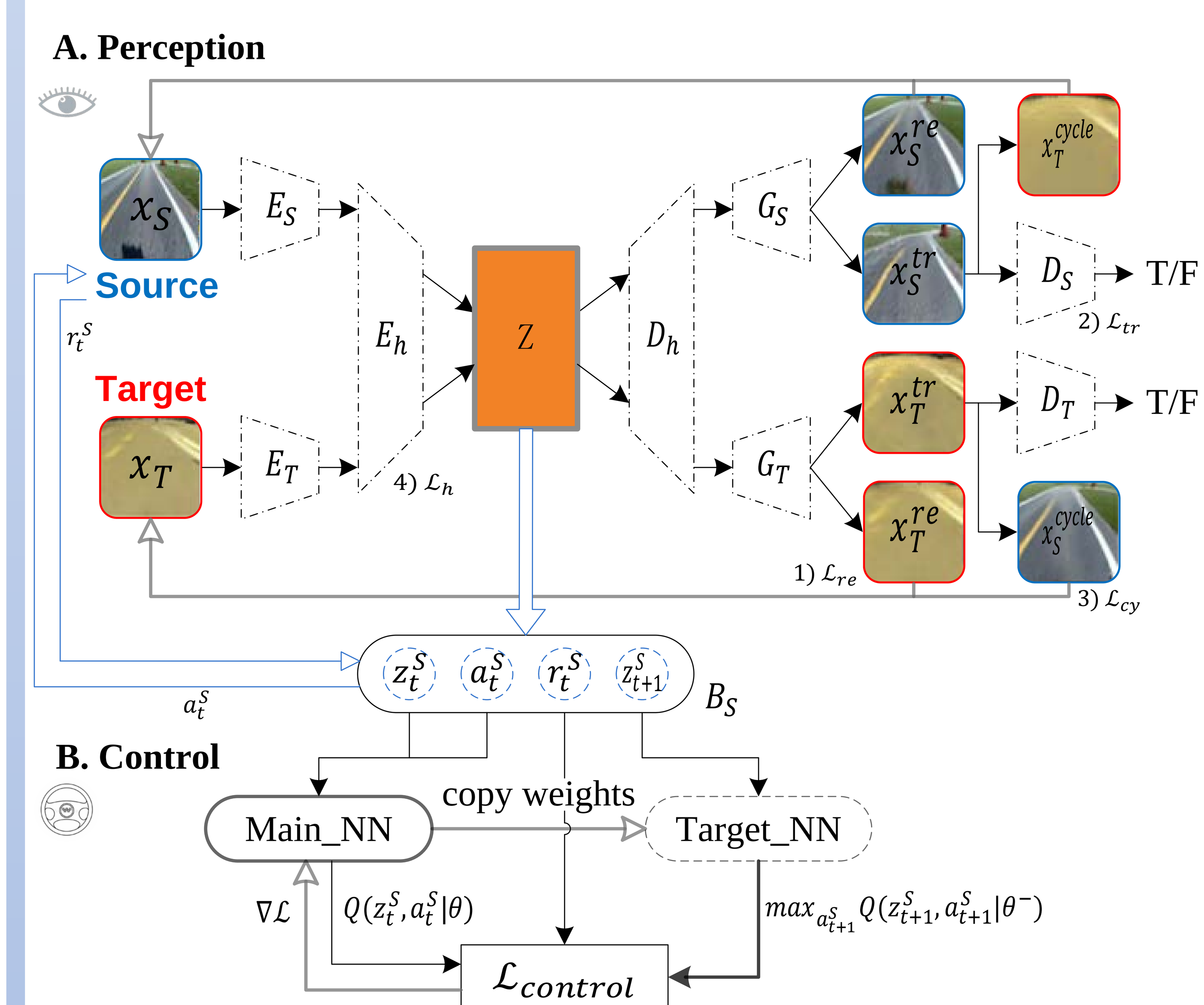
d) Weight sharing: $\mathcal{L}_h = \mathbb{E}_{x_S\sim P_S}\left[(E_h(E_S(x_S)) + \eta)^2\right] + \mathbb{E}_{x_T\sim P_T}\left[(E_h(E_T(x_T)) + \eta)^2\right]$

e) Perceptual loss: $\mathcal{L}_{perceptual} = \lambda_{re}\mathcal{L}_{re} + \lambda_{tr}\mathcal{L}_{tr} + \lambda_{cy}\mathcal{L}_{cy} + \lambda_h\mathcal{L}_h$

f) Future accumulated reward: $Q^{\pi}(z_t^S, a_t^S) = E[r_{t+1}^S + \gamma r_{t+2}^S + \gamma^2 r_{t+3}^S + \cdots | z_t^S, a_t^S]$

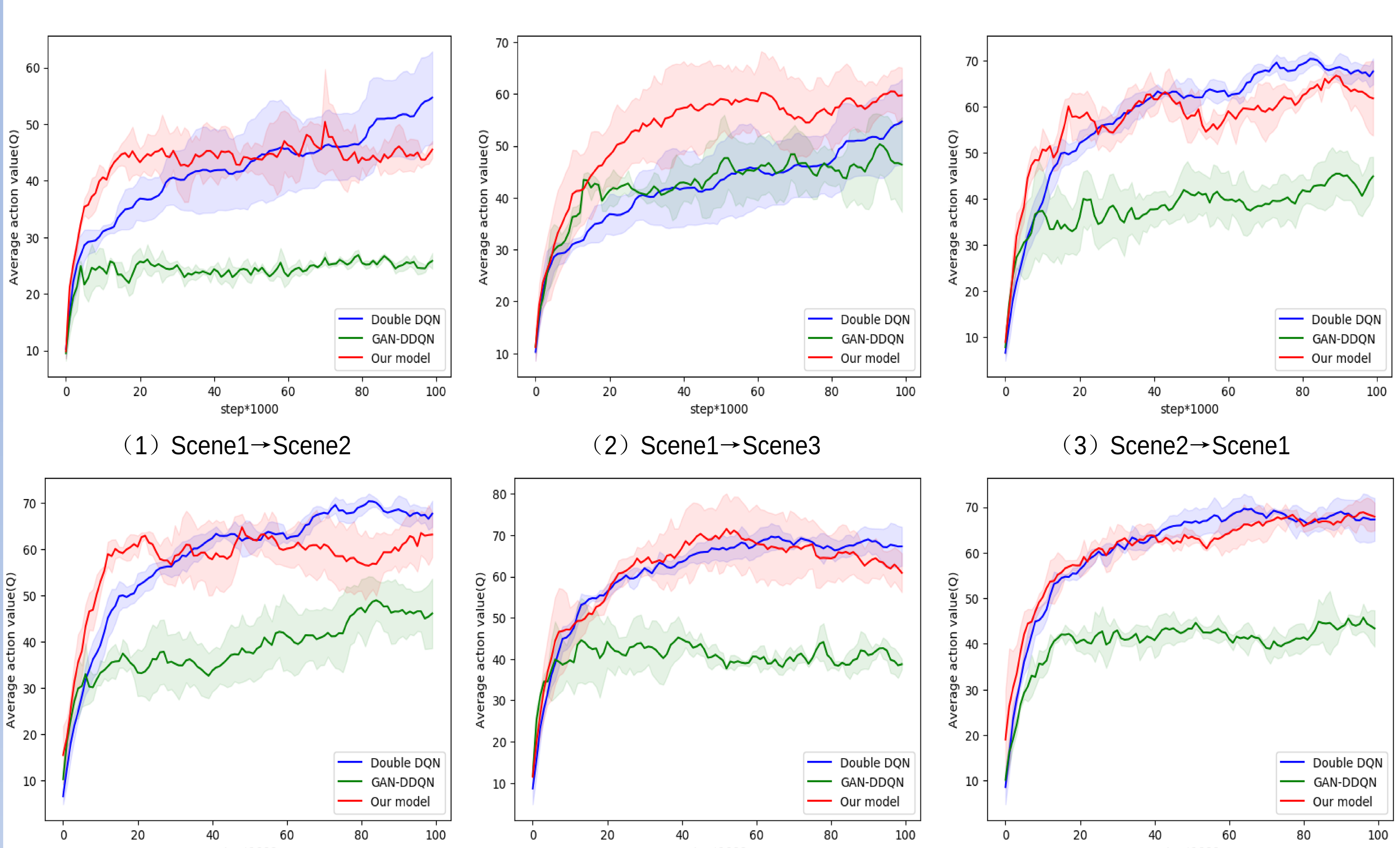
g) Control loss: $\mathcal{L}_{control} = E[(r_t^S + \gamma max_{a_{t+1}^S} Q(z_{t+1}^S, a_{t+1}^S|\theta^-) - Q(z_t^S, a_t^S|\theta))^2]$

Models



Results

TRAINING CURVES OF SIX TRANSFER TASKS



(1) Scene1→Scene2 (2) Scene1→Scene3 (3) Scene2→Scene1

(4) Scene2→Scene3 (5) Scene3→Scene1 (6) Scene3→Scene2

TRANSFER RESULTS BETWEEN DIFFERENT SCENES

Transfer Results		Double dqn		Gan-ddqn		Our model	
		T R	Laps%	T R	Laps%	T R	Laps%
Scene1→Scene2	Source	6962.2	100%	1742.2	23%	6081.7	100%
	Target	128.8	2.4%	353.1	6.2%	**5844.4**	**100%**
Scene1→Scene3	Source	7022.5	100%	2916.4	39%	6114.8	100%
	Target	261.4	5.4%	**4803.8**	**100%**	3769	100%
Scene2→Scene1	Source	6543.4	100%	2960.2	44%	7231.6	100%
	Target	951.8	20%	3391.4	91%	**5746.5**	**100%**
Scene2→Scene3	Source	7166.4	100%	94.8	2%	1870.3	38.5%
	Target	212.7	3%	5082.5	100%	**5195.8**	**100%**
Scene3→Scene1	Source	2363.4	50%	1100.7	20%	48591	100%
	Target	4603.1	73%	5447.7	79%	**7381.4**	**100%**
Scene3→Scene2	Source	2773.7	57%	125.6	3%	5625.5	100%
	Target	92.6	1.5%	736.8	12%	**6419.6**	**100%**