Goal-Aware Cross-Entropy for Multi-Target Reinforcement Learning

Kibeom Kim, Min Whoo Lee, Yoonsung Kim,

Je-Hwan Ryu, Minsu Lee, Byoung-Tak Zhang

Seoul National University

kbkim@bi.snu.ac.kr

For more realistic settings

- Need to handle multiple objects or destinations
  - Bring me a {*spoon*, *cup*, “*specific object*”}
  - Go to the {*kitchen*, *livingroom*, “*specific destination*”}

Human

AI Robot


Spoon


Cup


Keyboard

For more realistic settings

- Instruction-based multi-target task
  - It is still challenging task for RL
  - In existing studies, direct semantic understanding of the goal is necessary, but it is lacking.

For more realistic settings

- Instruction-based multi-target task
  - It is still challenging task for RL
  - *Targets* are possible goal candidates

For more realistic settings

- Instruction-based multi-target task
  - It is still challenging task for RL
  - *Targets* are possible goal candidates
  - The *goal* z may be selected among the targets, specified with a cue or an instruction
  - The instruction I^z is given randomly every episode, *"Bring me a spoon"*

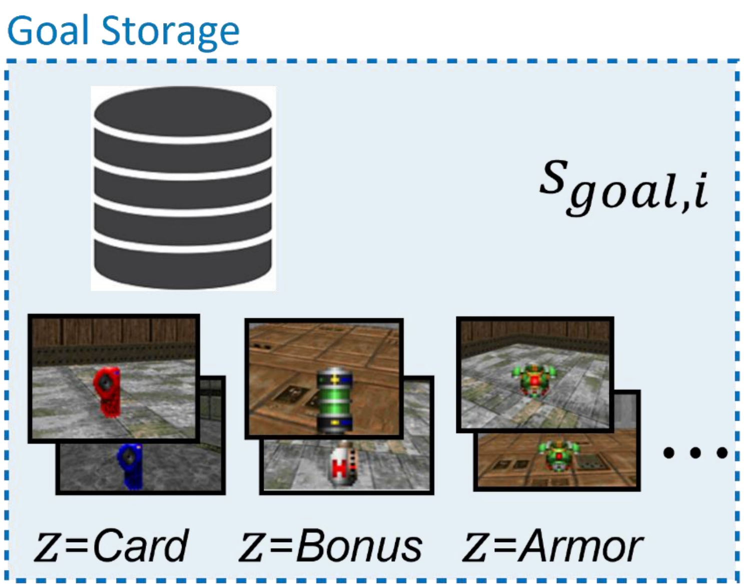
For more realistic settings

- Instruction-based multi-target task
  - It is still challenging task for RL
  - *Targets* are possible goal candidates
  - The *goal* z may be selected among the targets, specified with a cue or an instruction

- We propose a *Goal-Aware Cross-Entropy (GACE)* loss and *Goal-Discriminative Attention Networks (GDAN)* for multi-target reinforcement learning.

Collecting goal states

- Auto-labeled goal states for self-supervised learning
  - The agent actively gathers the goal states relying only on the instruction I^z and reward given by the environment.



If success at time t:

Store (S_t, embed(*Armor*)) in Goal Storage

Proposed methods

- Goal-Aware Cross-Entropy (GACE) loss
  - It trains the goal-discriminator that facilitates semantic understanding of goals alongside the policy
  - $s_{goal,i}$ is goal state, $\sigma(\cdot)$ is feature extractor and $d(\cdot)$ is goal-discriminator

$$\mathbf{e}_{s_{goal,i}} = \sigma(s_{goal,i})$$

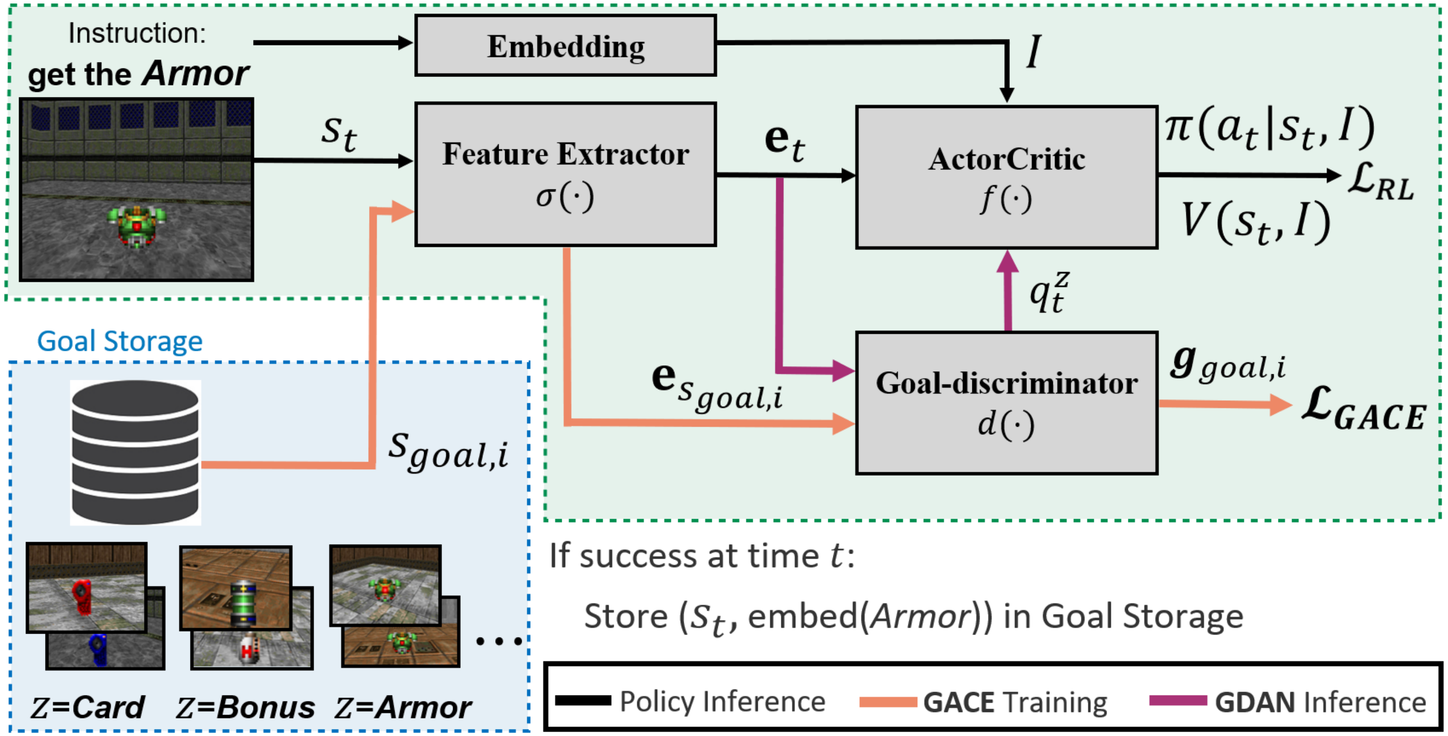
$$\mathbf{g}_{goal,i} = d(\mathbf{e}_{s_{goal,i}})$$

  - z_i is the automatic label corresponding to state $\mathbf{g}_{goal,i}$
  - Then, GACE loss is

$$\mathcal{L}_{GACE} = -\sum_{i=0}^{M-1} one\_hot(z_i) \cdot \log(\mathbf{g}_{goal,i})$$

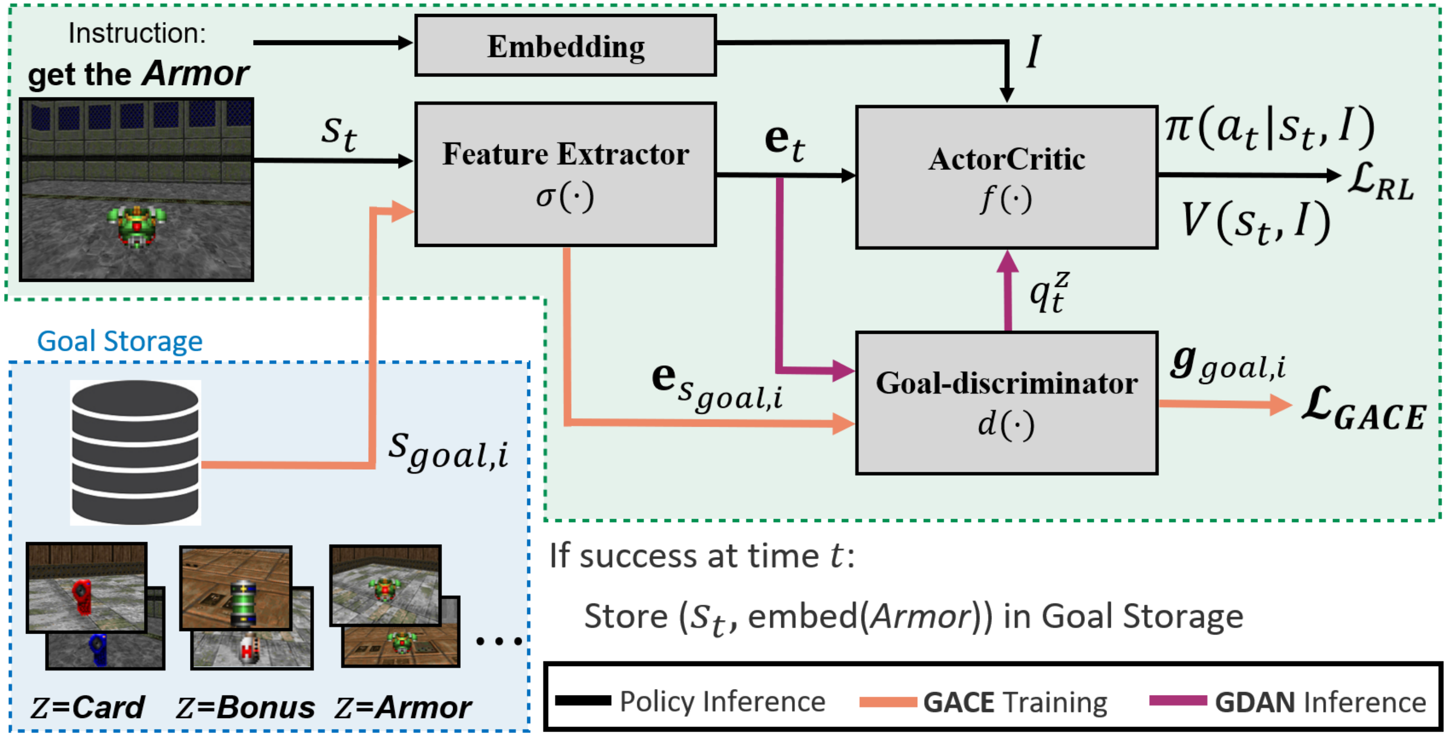
Overview of architecture

- Goal-Aware Cross-Entropy (GACE) loss



Overview of architecture

- Goal-Aware Cross-Entropy (GACE) loss

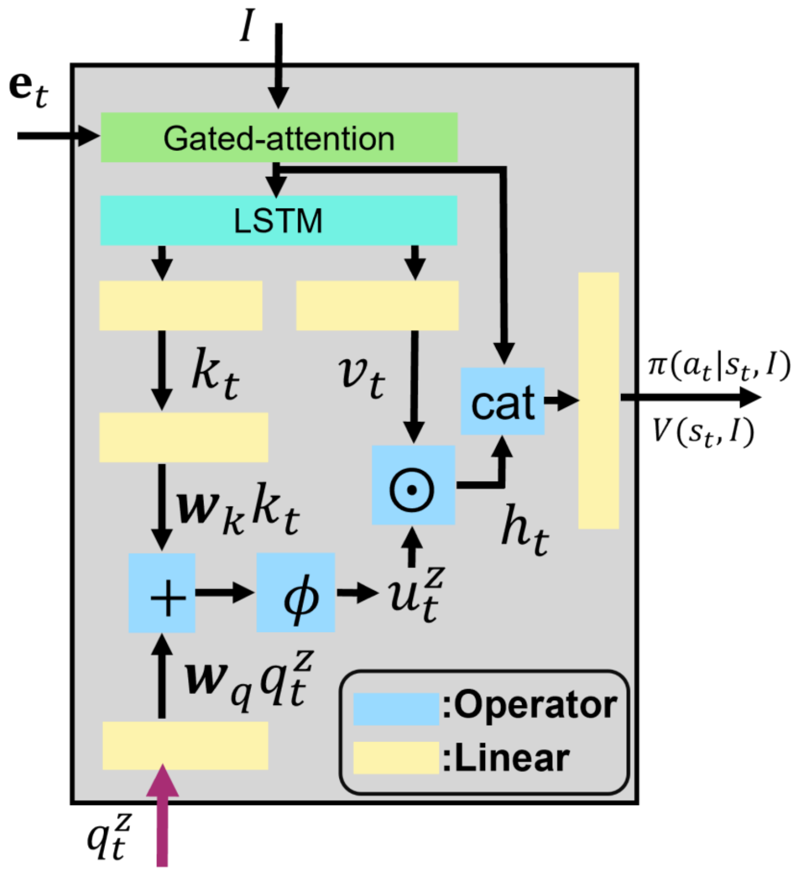


  - The **GACE loss** makes the goal-discriminator become goal-aware without external supervision.
  - Such goal-awareness is advantageous for sample-efficiency and generalization in multi-target environments.

Proposed methods

- Goal-Discriminative Attention Networks (GDAN)
  - Goal-relevant query q_t^z from goal-discriminator
  - The *key* k_t and *value* v_t from encoded state in the ActorCritic $f(\cdot)$

$$u_t^z = \phi(\mathbf{W_q} q_t^z + \mathbf{W_k} k_t)$$
$$h_t = v_t \odot u_t^z$$



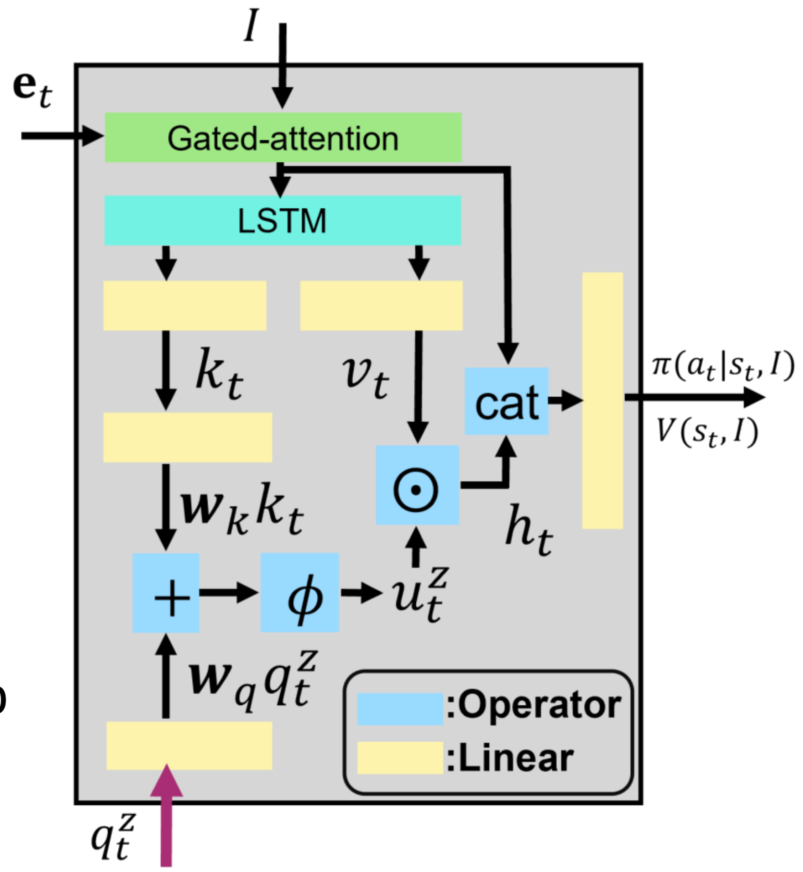
<Attention in ActorCritic $f(\cdot)$>

Proposed methods

- Goal-Discriminative Attention Networks (GDAN)
  - Goal-relevant query q_t^z from goal-discriminator
  - The *key* k_t and *value* v_t from encoded state in the ActorCritic $f(\cdot)$

$$u_t^z = \phi(\mathbf{W_q} q_t^z + \mathbf{W_k} k_t)$$
$$h_t = v_t \odot u_t^z$$

  - It makes the agent to selectively allocate attention for goal-directed actions
  - effectively utilize the discriminator to enhance the performance and efficiency

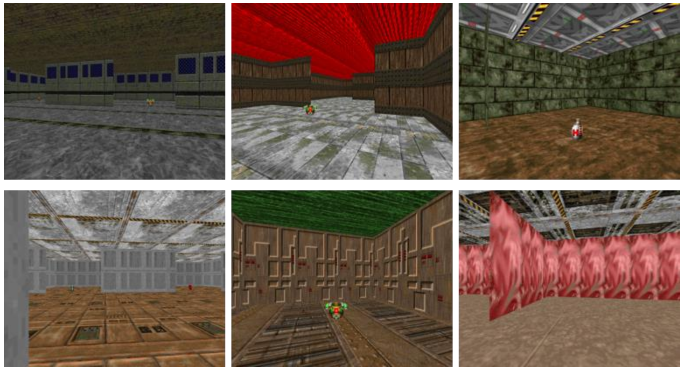


<Attention in ActorCritic $f(\cdot)$>

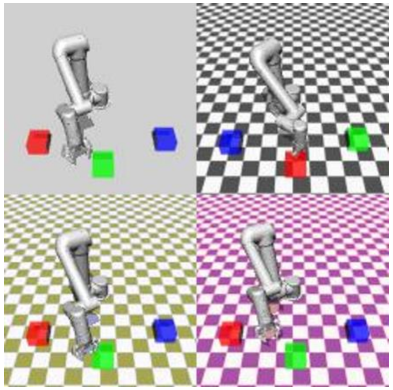
Environments

- Multi-target environments
  - The object positions are randomly shuffled to learn discriminability.
  - Background is also randomly selected to evaluate generalization.

- Visual navigation tasks
  - First-person view
  - 4 classes 8 objects
  - “Get the Armor / Bonus / Card / …”

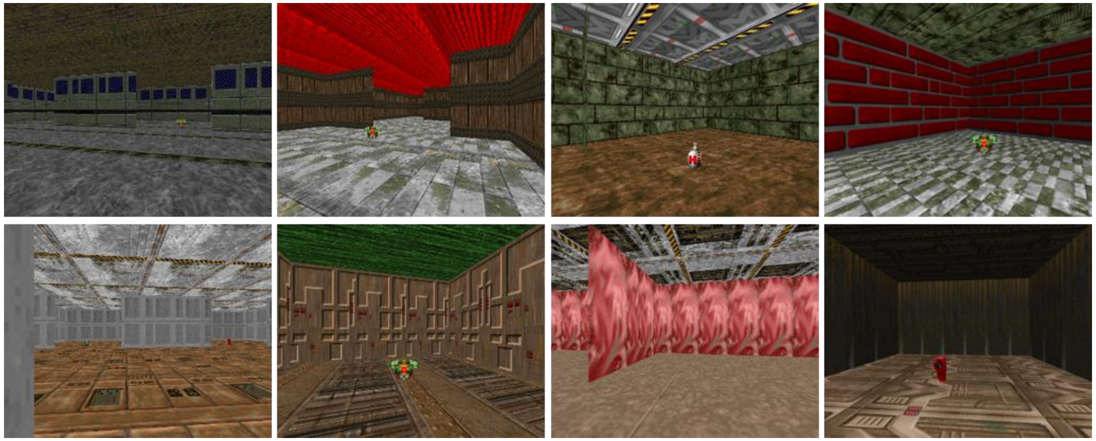


- Robot arm manipulation tasks
  - Fixed third-person view
  - 3 or 5 objects for each task
  - “Reach the red/blue/green box”

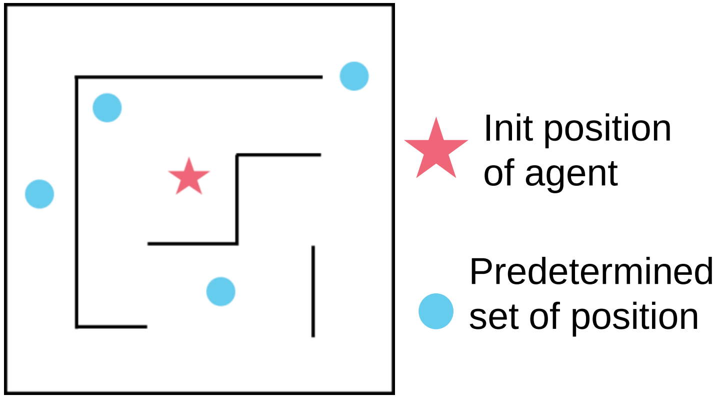


Environment

- Visual navigation
  - **V1**
  - **V2 seen, unseen**
  - **V3**
  - **V4 seen, unseen**



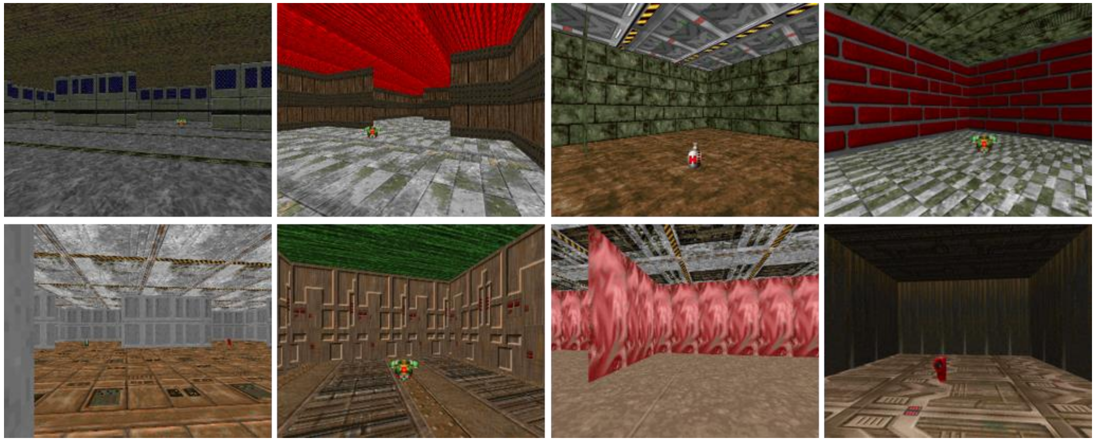
<Samples of used textures>



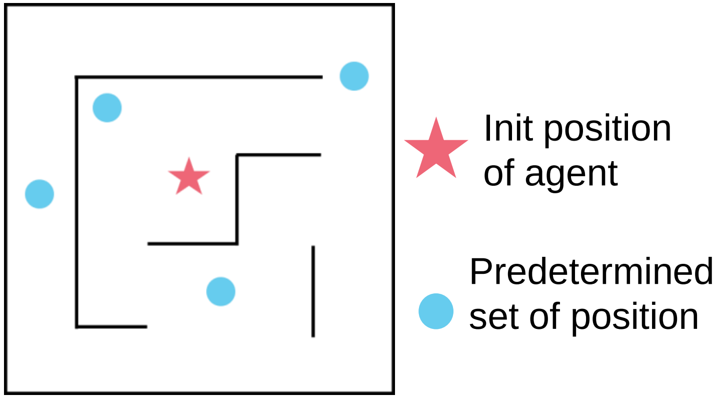
<Top-down view of V3,V4>

Environment

- Visual navigation
  - **V1**: default navigation task
    closed rectangular room with no walls
  - **V2 seen, unseen**: to evaluate generalization
    added textures in V1 setting
  - **V3**: more complex than V1, additional walls
  - **V4 seen, unseen**: added textures in V3 setting



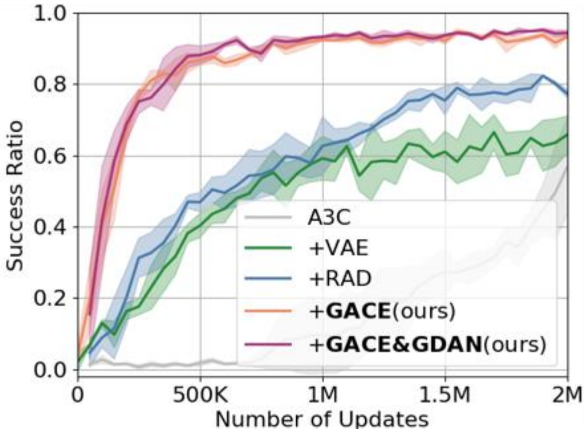
<Samples of used textures>



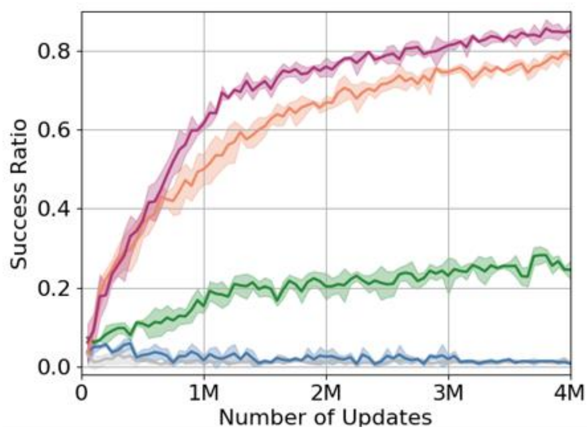
<Top-down view of V3,V4>

Experiments

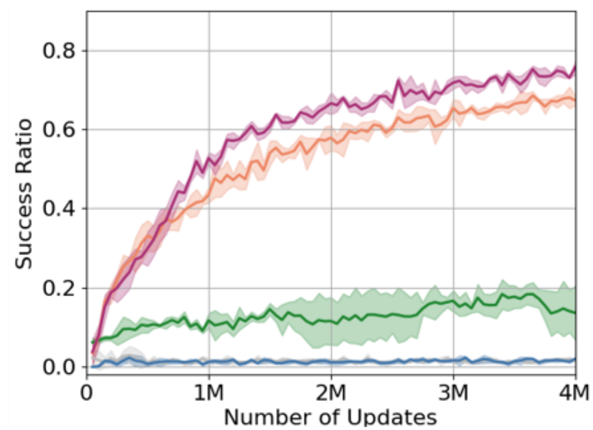
- Visual navigation task



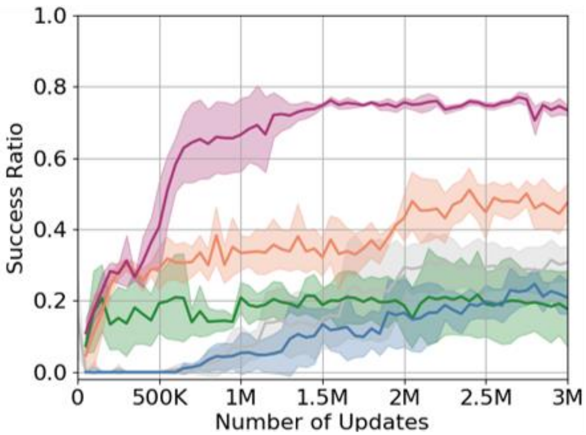
<V1>



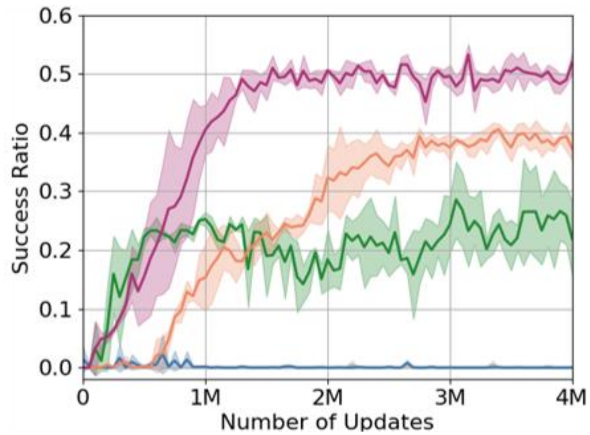
<V2 Seen>



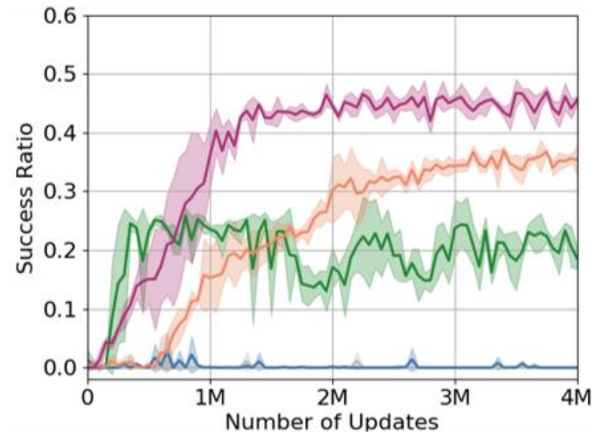
<V2 Unseen>



<V3>



<V4 Seen>



<V4 Unseen>

Experiments

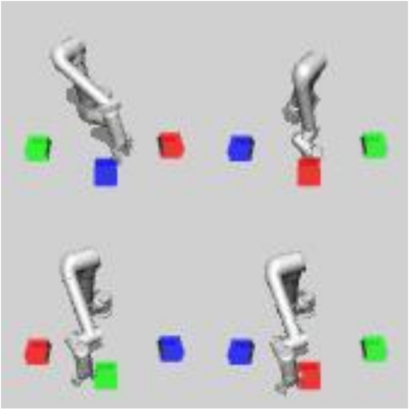
- Sample-efficiency metric for V1 task

Table 1: Success ratio (SR) and sample efficiency metrics in visual navigation task **V1**. SRR (lower the better) and SEI (higher the better) are measured with A3C as a reference. "Number of Updates" indicates the number of updates required to reach the reference performance.

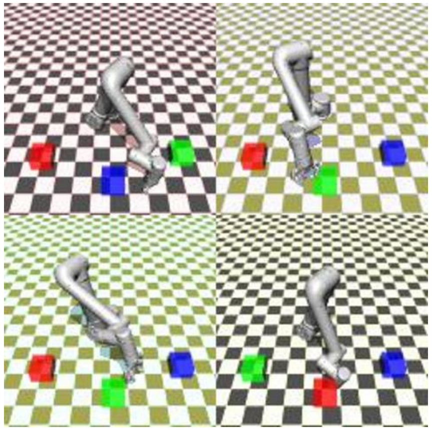
Algorithm	SR of **V1** (%)	Number of Updates	SRR (%)	SEI (%)
A3C	56.55 ± 13.8	2M	100	-
+VAE	67.89 ± 3.5	810,086	40.50	146.89
+RAD	82.14 ± 2.3	703,574	35.18	184.26
+**GACE** (ours)	$\mathbf{94.97 \pm 0.7}$	163,602	8.18	1122.48
+**GACE** & **GDAN** (ours)	$\mathbf{95.6 \pm 0.64}$	110,930	**5.55**	**1702.94**

Environment

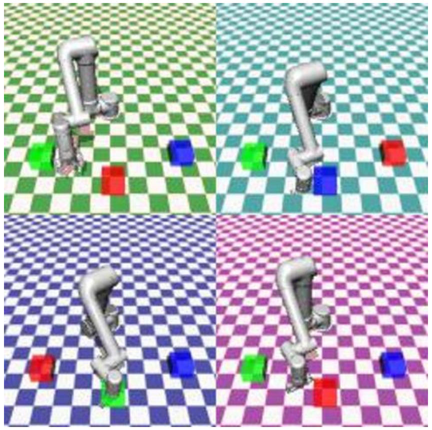
- Robot arm manipulation task
  - **R1**
  - **R2 seen, unseen**
  - **R3**



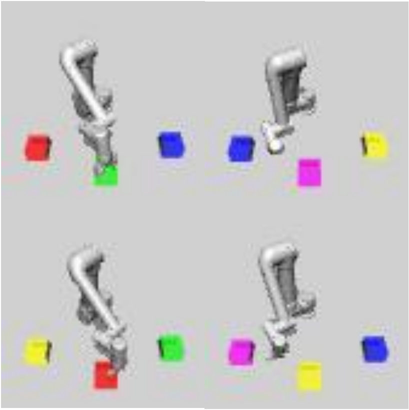
<R1>



<R2 seen>



<R2 unseen>



<R3>

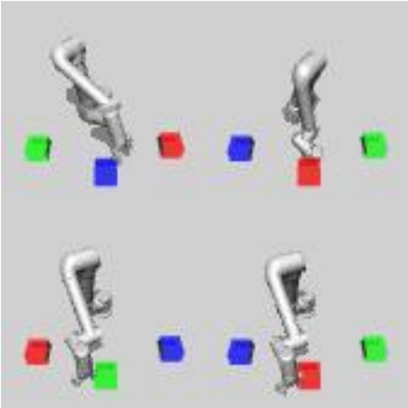
Environment

- Robot arm manipulation task
  - **R1:** default manipulation task

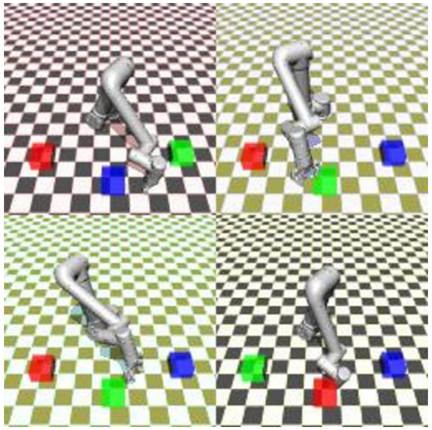
    red/green/blue box are randomly shuffled
  - **R2 seen, unseen:** to evaluate generalization

    added checkered background
  - **R3:** to evaluate scalability with more targets

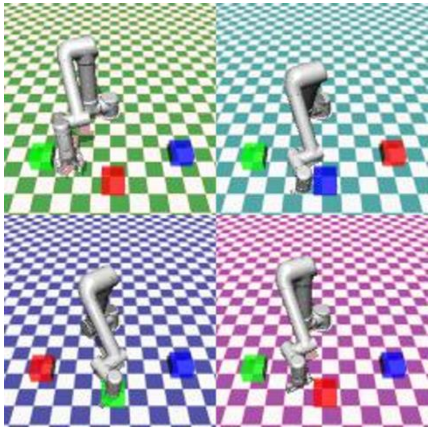
    + yellow/pink box in R1 setting



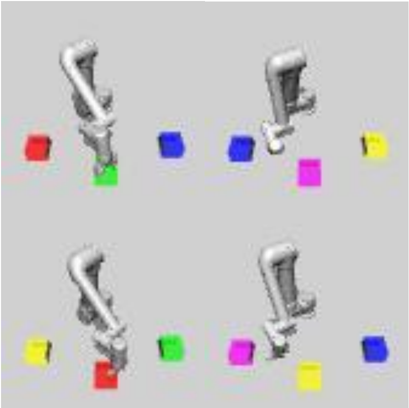
<R1>



<R2 seen>



<R2 unseen>



<R3>

Experiments

- Robot arm manipulation task

Table 2: Success ratio (SR) in robot arm manipulation tasks.

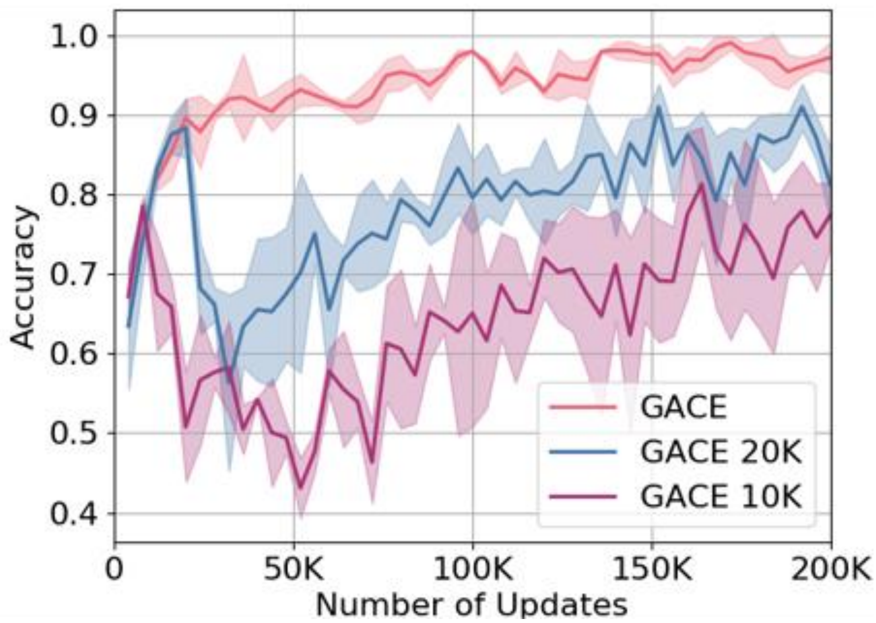
Algorithm	SR of **R1** (%)	SR of **R2** Seen (%)	SR of **R2** Unseen (%)	SR of **R3** (%)
SAC	63.1 ± 6.9	60.5 ± 5.7	53.4 ± 6.9	61.7 ± 5.4
+AE	67.2 ± 5.0	72.8 ± 5.9	59.4 ± 5.5	62.3 ± 5.1
+CURL	67.9 ± 7.3	74.5 ± 9.2	36.6 ± 3.4	64.7 ± 4.0
+GACE	84.7 ± 10.0	75.0 ± 8.9	63.0 ± 9.0	$\mathbf{79.3 \pm 8.9}$
+GACE&GDAN	$\mathbf{89.3 \pm 4.2}$	$\mathbf{78.2 \pm 8.7}$	$\mathbf{73.3 \pm 5.8}$	$\mathbf{79.6 \pm 8.4}$

Table 3: Sample efficiency metrics for R1 task. SR is reference performance of R1 task.

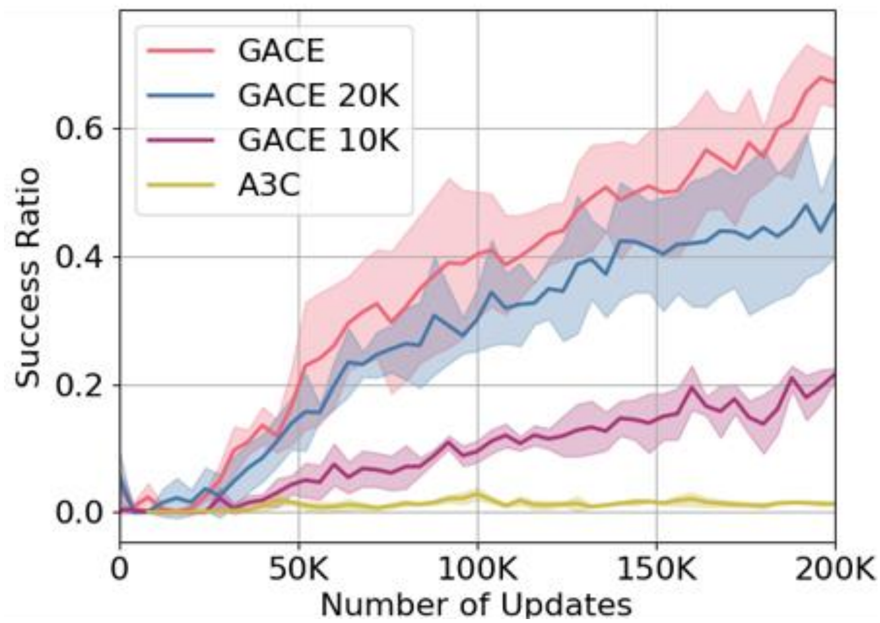
Algorithm	SR (%)	Number of Updates	SRR (%)	SEI (%)
SAC	63.1	314,797	100	-
+AE		230,339	73.17	36.67
+CURL		142,480	45.26	120.94
+GACE (ours)		53,774	**17.08**	**485.41**
+GACE&GDAN (ours)		63,140	20.06	398.57

Analysis

- Effectiveness of GACE



<Goal discriminator accuracy>

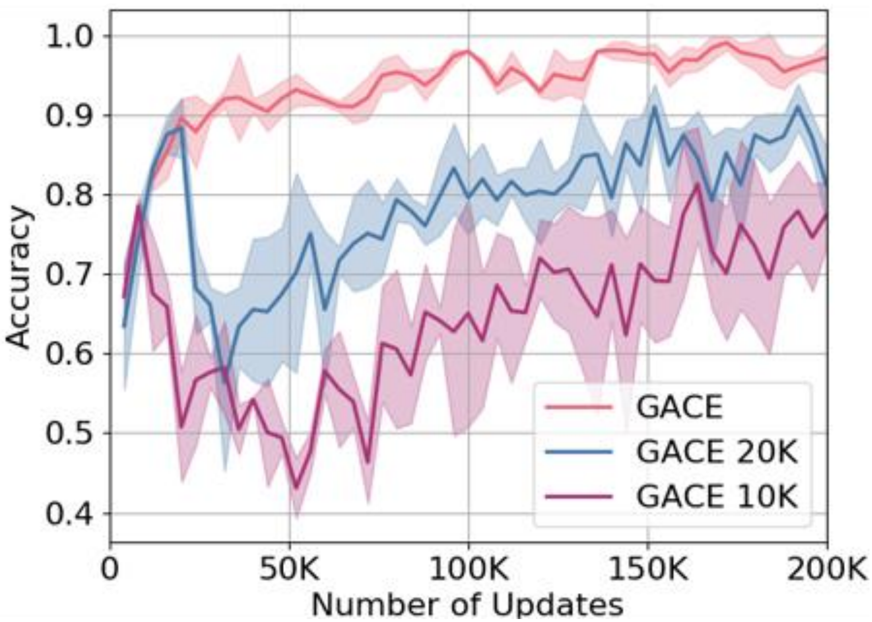


<Learning curve in V1 task>

The goal-discriminator weights are unfrozen (red),
frozen at 10K (purple) and 20K (blue) updates.

Analysis

Effectiveness of GACE



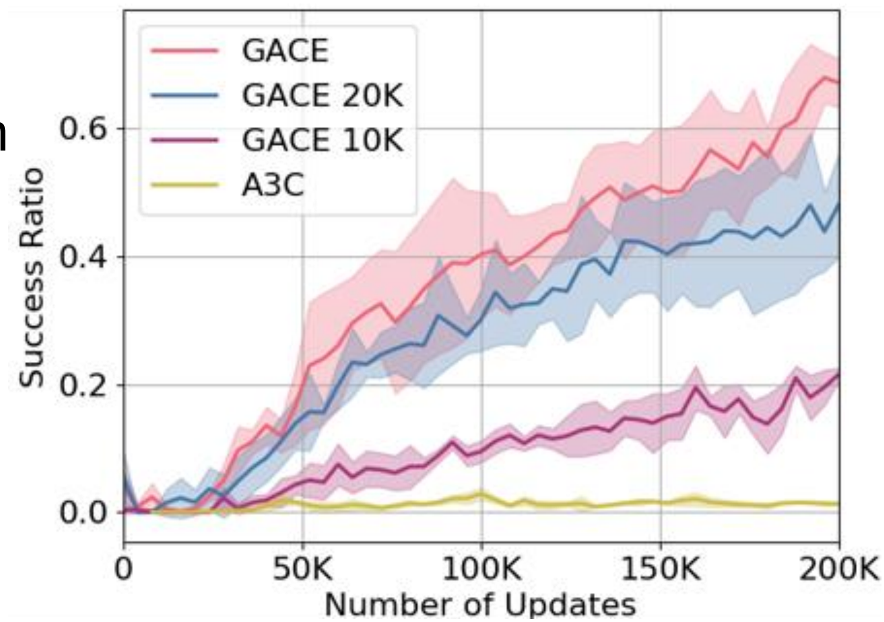
<Goal discriminator accuracy>

- Although the GACE loss (frozen weights) does not further contribute to learning, the discriminator accuracy improves only by updating the policy.
- This indicates that throughout the training, the agent gradually develops a feature extractor σ(·) that can discriminate targets.

The goal-discriminator weights are unfrozen (red), frozen at 10K (purple) and 20K (blue) updates.

Analysis

- Effectiveness of GACE
  - Even when the agent is trained with the GACE only temporarily, the learning curve is steeper than that with vanilla A3C.
  - Consequently, learning GACE loss has positive influence on policy performance than learning solely with policy updates.



<Learning curve in V1 task>

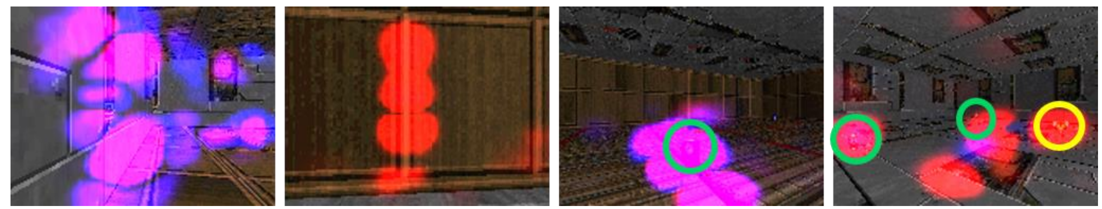
The goal-discriminator weights are unfrozen (red),
frozen at 10K (purple) and 20K (blue) updates.

Analysis

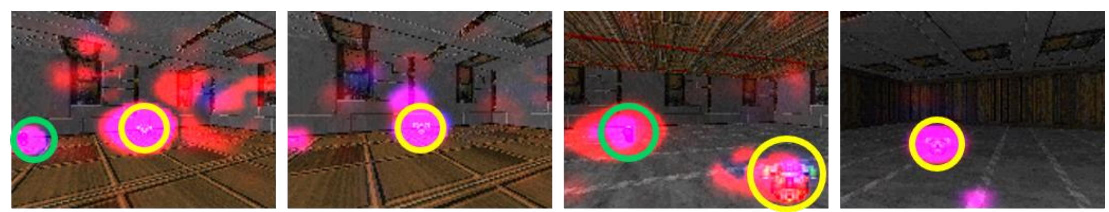
- Visual interpretation using saliency map

: non-goal

: goal



<A3C>



<GACE>

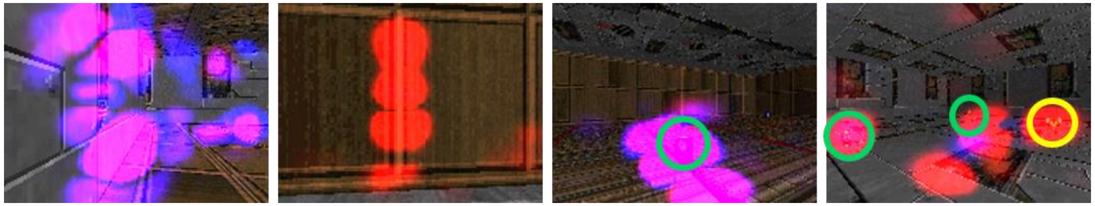
<GACE & GDAN>

Analysis

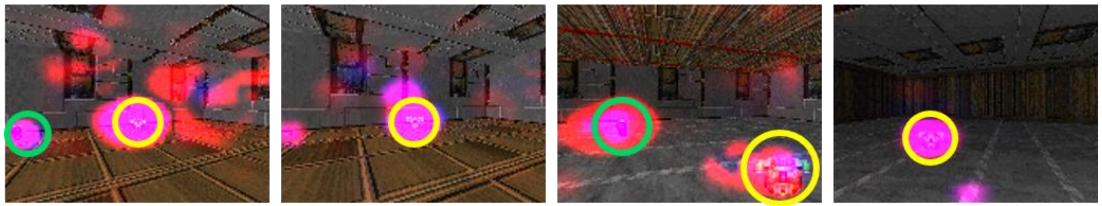
- Visual interpretation using saliency map
  - The agent is overly sensitive to edges in the background in *A3C*.

: non-goal

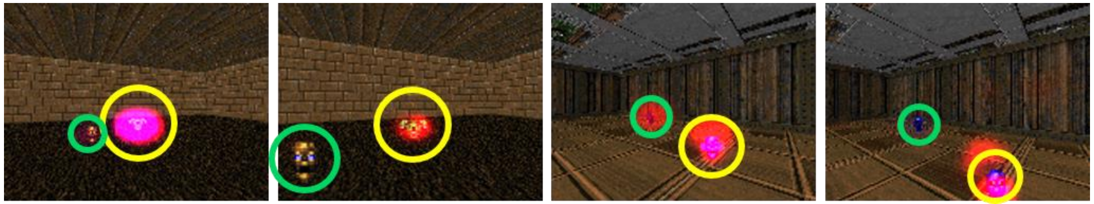
: goal



<A3C>



<GACE>



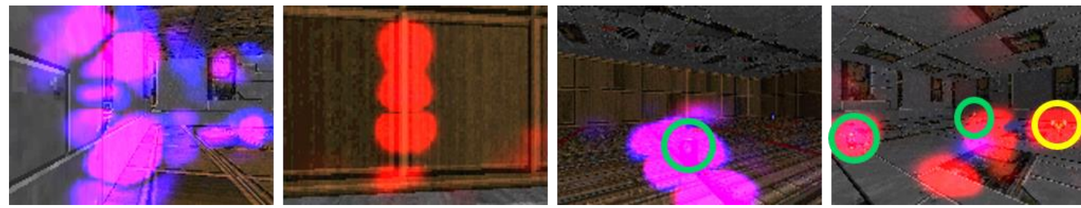
<GACE & GDAN>

Analysis

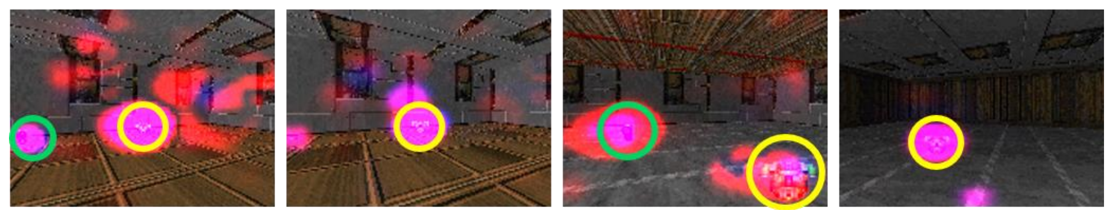
: non-goal

: goal

- Visual interpretation using saliency map
  - The agent is overly sensitive to edges in the background in *A3C*.
  - All goals and non-goals are detected successfully in *GACE.*



<A3C>



<GACE>

<GACE & GDAN>

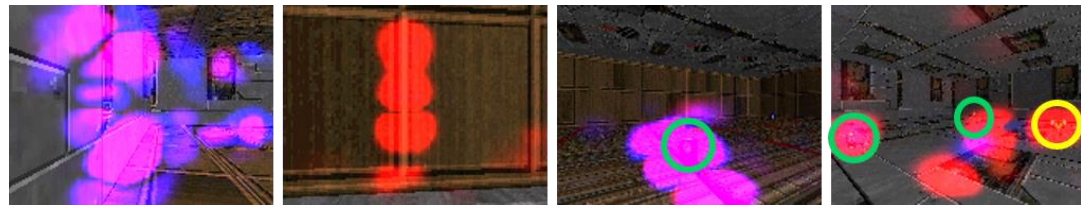
Analysis

- Visual interpretation using saliency map

: non-goal

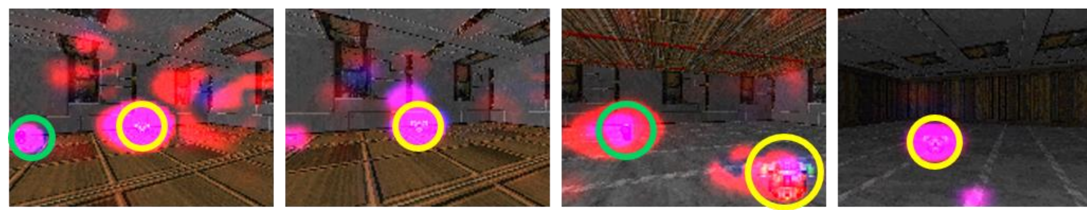
: goal

  - The agent is overly sensitive to edges in the background in *A3C*.



<A3C>

  - All goals and non-goals are detected successfully in *GACE.*



<GACE>

  - The agent shows sensitive reaction only to goal in *GACE&GDAN.*

<GACE & GDAN>

Conclusion

- We propose GACE loss and GDAN for multi-target RL.
  - It learns goal states in a self-supervised manner using a reward and instruction.
  - It promotes a goal-focused behavior.
  - Our methods achieve state-of-the-art sample-efficiency and generalization in multi-target environments.