논문 제목 : VARIATIONAL DISCRIMINATOR BOTTLENECK: IMPROVING IMITATION LEARNING, INVERSE RL, AND GANS BY CONSTRAINING INFORMATION FLOW

1. Introduction
    a. 기존 GAN 이 학습을 잘하기 위해서 Generator 와 Discriminator 간의 학습 균형이 맞아야 하는데 Discriminator 만 정확도가 높아지며 Uninformative Gradiant 만 나온다.
    b. 따라서 이런 문제를 해결하기 위해서(Discriminator 의 accuracy 를 제약) Discriminator 의 Internal representation 과 input distribution 간에 Mutual information 을 제약시킴으로써 Discriminator 와 Generator 간의 균등한 학습을 할 수 있도록 Variational Discriminator Bottleneck(VDB)->Adaptive Stochastic Regularization 기법을 제안한다.
    c. 실험:
        i. Adversarial Imitation Learning, Inverse Reinforcement Learning 평가, 이미지 생성 평가
    d. Contribution
        i. VDB 기법 제안
        ii. 여러 테스크에 사용가능(GAN 들어가는 곳은 모두 사용가능)
        iii.

Figure 1: Our method is general and can be applied to a broad range of adversarial learning tasks. **Left:** Motion imitation with adversarial imitation learning. **Middle:** Image generation. **Right:** Learning transferable reward functions through adversarial inverse reinforcement learning.

2. Related Work
    a. GAN
        i. GAN 기법이 이미지 생성분야에서 좋은 성능을 냄. (GAIL 과 같이 IRL 에 Reward inference 하는데도 사용)
        ii. 하지만 Discriminator 와 Generator 간에 균형적으로 학습을 해야하는데 Discriminator 만 잘 학습되면 Generator 가 Gradient 를 학습할 때 Uninformative Gradient 가 발생
        iii. 이런 문제를 mitigate 하기 위해 Regularizer 를 제안(**stability**, **Convergence** 를 향상)

1. **Gradient Penalties**, **Reconstruction loss** 등이 있음

iv. 이 논문에서도 비슷하게 Regularization 을 사용(Discriminator 로부터 Information bottleneck 을 적용)

b. Variational Discriminator Bottleneck(VDB)

i. **Information Bottleneck** 을 기반으로 하고 있다.

ii. 이 기술은 Input 과의 상호정보량을 최소화 시키기 위해서 Internal Representation 을 정규화(직관적으로 압축된 Representation 일반화를 향상시키는데, 그 방법은 주어진 **Input Feature 에 관련이 없는 Distractor 를 무시하는 것**)

iii. Information bottleneck 은 Deep Neural Network 에서 Feature 를 압축하는 것과 같은 효과

3. Preliminaries

a. Variational Information Bottleneck(Supervised Learning 관점에서)

i. DataSet $\{x_i, y_i\}$, with features x_i and labels y_i

ii. standard maximum likelihood estimate $q(y_i|x_i)$

$$\min_{q} \quad \mathbb{E}_{\mathbf{x},\mathbf{y}\sim p(\mathbf{x},\mathbf{y})}\left[-\log q(\mathbf{y}|\mathbf{x})\right].$$

iii.

iv. 하지만 이식은 Overfitting 될 확률이 많다. 따라서 Mutual Information 을 이용한 Regularizing 기법인 Information bottleneck 이란 기법을 제안

v. encoder $E(z|x)$ that maps the features x to a latent distribution over Z

vi. upper bound I_c on the mutual information between the encoding and the original features

vii. I(X, Z)(Mutual Information).

viii. objective J(q, E)

ix.

$$J(q,E) = \min_{q,E} \quad \mathbb{E}_{\mathbf{x},\mathbf{y}\sim p(\mathbf{x},\mathbf{y})}\left[\mathbb{E}_{\mathbf{z}\sim E(\mathbf{z}|\mathbf{x})}\left[-\log q(\mathbf{y}|\mathbf{z})\right]\right]$$
$$\text{s.t.} \quad I(X,Z) \leq I_c.$$

x.

xi. Mutual Information 은 하나의 확률변수를 발견했을 때 또 다른 확률변수에서 얻을 수 있는 정보량

$$
\begin{aligned}
I(X,Z) &= \int p(x,z)\log\frac{p(x,z)}{p(x)p(z)}dxdz \\
&= \int p(x)\frac{p(x,z)}{p(x)}\log\frac{p(x,z)}{p(x)p(z)}dxdz \\
&= \int p(x)E(z|x)\log\frac{E(z|x)}{p(z)}dxdz
\end{aligned}
$$

xii.

xiii. p(x) is the distribution given by the dataset

xiv. Marginal Distribution:

1. p(z) = E(z|x) p(x) dx 하는게 어려운 일이므로 이것을 근사화하는 Marginal r(z) 도입

xv. p(z)와 r(z)의 항상 참인명제 KL [p(z)||r(z)] ≥ 0 를 이용한 Mutual Information 의 의 Upper bound 를 구함

$$
\begin{aligned}
KL[p(z)\|r(z)] &= \int p(z)\log\frac{p(z)}{r(z)}dz \\
&= \int p(z)\log p(z)dz - \int p(z)\log r(z)dz \\
&\geq 0
\end{aligned}
$$

$$
\Leftrightarrow \int p(z)\log p(z)dz \geq \int p(z)\log r(z)dz
$$

xvi.

$$
\begin{aligned}
I(X,Z) &= \int p(x)E(z|x)\log\frac{E(z|x)}{p(z)}dxdz \\
&\leq \int p(x)E(z|x)\log\frac{E(z|x)}{r(z)}dxdz \\
&= \mathbb{E}_{x\sim p(x)}\Big[KL[E(z|x)\|r(z)]\Big].
\end{aligned}
$$

xvii.

xviii. 그리고 구하려고 했던 Regularized Object 에 Upper Bound 식으로 변환

xix. J(q, E) ≥ J(q, E) -> 근사화

$$
\begin{aligned}
\tilde{J}(q,E) = \min_{q,E} \quad & \mathbb{E}_{\mathbf{x},\mathbf{y}\sim p(\mathbf{x},\mathbf{y})}\left[\mathbb{E}_{\mathbf{z}\sim E(\mathbf{z}|\mathbf{x})}\left[-\log q(\mathbf{y}|\mathbf{z})\right]\right] \\
\text{s.t.} \quad & \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}\left[\mathrm{KL}\left[E(\mathbf{z}|\mathbf{x})||r(\mathbf{z})\right]\right] \leq I_c.
\end{aligned}
$$

xx.

xxi. Unconstrainted Optimization 으로 위에 문제를 접근하기 위해서 Lagrangian 형태로 변환하여 Unconstrainted 문제로 변환(beta 는 상관계수)

xxii. $$\min_{q,E} \quad \mathbb{E}_{\mathbf{x},\mathbf{y}\sim p(\mathbf{x},\mathbf{y})}\left[\mathbb{E}_{\mathbf{z}\sim E(\mathbf{z}|\mathbf{x})}\left[-\log q(\mathbf{y}|\mathbf{z})\right]\right] + \beta\left(\mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}\left[\mathrm{KL}\left[E(\mathbf{z}|\mathbf{x})||r(\mathbf{z})\right]\right] - I_c\right).$$

4. Variational Discriminator Bottleneck
   a. 기본 GAN Network
      i. $$\max_G \min_D \quad \mathbb{E}_{\mathbf{x}\sim p^*(\mathbf{x})}\left[-\log\left(D(\mathbf{x})\right)\right] + \mathbb{E}_{\mathbf{x}\sim G(\mathbf{x})}\left[-\log\left(1-D(\mathbf{x})\right)\right].$$
   b. Variational information bottleneck 추가
      i. $$J(D,E) = \min_{D,E} \quad \mathbb{E}_{x\sim p^*(\mathbf{x})}\left[\mathbb{E}_{\mathbf{z}\sim E(\mathbf{z}|\mathbf{x})}\left[-\log\left(D(\mathbf{z})\right)\right]\right] + \mathbb{E}_{\mathbf{x}\sim G(\mathbf{x})}\left[\mathbb{E}_{\mathbf{z}\sim E(\mathbf{z}|\mathbf{x})}\left[-\log\left(1-D(\mathbf{z})\right)\right]\right]$$
      $$\text{s.t.} \quad \mathbb{E}_{\mathbf{x}\sim\tilde{p}(\mathbf{x})}\left[\mathrm{KL}\left[E(\mathbf{z}|\mathbf{x})||r(\mathbf{z})\right]\right] \leq I_c,$$
      ii. with $\tilde{p} = \frac{1}{2}p^* + \frac{1}{2}G$ being a mixture of the target distribution and the generator.
      iii. Mixture Distribution 은 학습이 잘 되지 않는 초반 G 에 잘 일어날 수 있는 High Variance 를 잡기 위해서 사용
   c. 위에 문제를 Variational Discriminator Bottleneck 이라 명하고, Object 를 Optimize 한다
      i. $$J(D,E) = \min_{D,E}\max_{\beta\geq 0} \quad \mathbb{E}_{\mathbf{x}\sim p^*(\mathbf{x})}\left[\mathbb{E}_{\mathbf{z}\sim E(\mathbf{z}|\mathbf{x})}\left[-\log\left(D(\mathbf{z})\right)\right]\right] + \mathbb{E}_{\mathbf{x}\sim G(\mathbf{x})}\left[\mathbb{E}_{\mathbf{z}\sim E(\mathbf{z}|\mathbf{x})}\left[-\log\left(1-D(\mathbf{z})\right)\right]\right]$$
      $$+\beta\left(\mathbb{E}_{\mathbf{x}\sim\tilde{p}(\mathbf{x})}\left[\mathrm{KL}\left[E(\mathbf{z}|\mathbf{x})||r(\mathbf{z})\right]\right] - I_c\right).$$
   d. 위에 문제를 Dual Gradient 문제로 풀면 D, E, B 에 대한 Update 는
      i. $$D, E \leftarrow \arg\min_{D,E} \mathcal{L}(D,E,\beta)$$
      $$\beta \leftarrow \max\left(0,\ \beta + \alpha_\beta\left(\mathbb{E}_{\mathbf{x}\sim\tilde{p}(\mathbf{x})}\left[\mathrm{KL}\left[E(\mathbf{z}|\mathbf{x})||r(\mathbf{z})\right]\right] - I_c\right)\right),$$
      ii. where $\mathcal{L}(D,E,\beta)$ is the Lagrangian
      $$\mathcal{L}(D,E,\beta) = \mathbb{E}_{\mathbf{x}\sim p^*(\mathbf{x})}\left[\mathbb{E}_{\mathbf{z}\sim E(\mathbf{z}|\mathbf{x})}\left[-\log\left(D(\mathbf{z})\right)\right]\right] + \mathbb{E}_{\mathbf{x}\sim G(\mathbf{x})}\left[\mathbb{E}_{\mathbf{z}\sim E(\mathbf{z}|\mathbf{x})}\left[-\log\left(1-D(\mathbf{z})\right)\right]\right]$$
      $$+\beta\left(\mathbb{E}_{\mathbf{x}\sim\tilde{p}(\mathbf{x})}\left[\mathrm{KL}\left[E(\mathbf{z}|\mathbf{x})||r(\mathbf{z})\right]\right] - I_c\right),$$
      iii. α_β is the step size
      iv. D, E 를 one step gradient 를 하고난 후 B 를 업데이트
   e. VDB+GAN -> Variational Generative Adversarial Network(VGAN)이라함
   f. 나머지

논문의 실험에서는 r을 정규분포로 설정한다. $(r(z) = N(0, I))$

Encoder는 mean μ_E, diagonal covariance matrix $\Sigma_E(x)$의 정규분포로 정의한다.
$E(z|x) = N(\mu_E(x), \Sigma_E(x)$

Generator의 목적함수에는 Z에 대한 expectation이 아니라 $\mu_E(x)$를 이용한 근사식을 사용하여 실험에서 충분한 성능을 냈다.

$$\max_G \mathbb{E}_{x\sim G(x)}[-\log(1 - D(\mu_E(x)))].$$

Discriminator는 sigmoid를 activation으로 사용하는 single linear unit으로 모델링되었다.

$$D(z) = \sigma(w_D^T z + b_D), \text{ with weights } w_D \text{ and bias } b_D.$$

i.

ii. r = Standard Gaussian

g. 최종

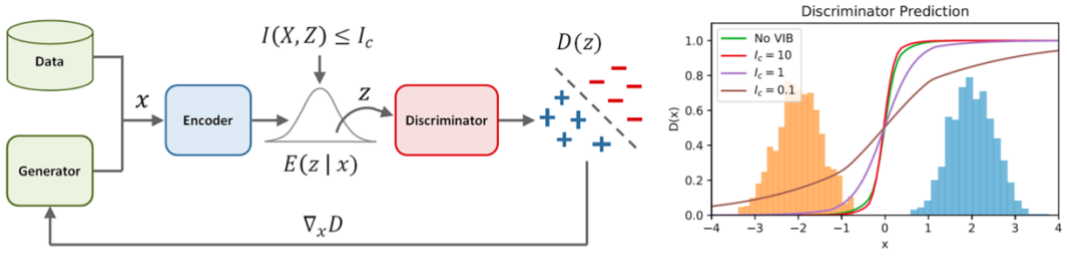


Figure 2: **Left:** Overview of the variational discriminator bottleneck. The encoder first maps samples $\mathbf{x}$ to a latent distribution $E(\mathbf{z}|\mathbf{x})$. The discriminator is then trained to classify samples $\mathbf{z}$ from the latent distribution. An information bottleneck $I(X, Z) \leq I_c$ is applied to Z. **Right:** Visualization of discriminators trained to differentiate two Gaussians with different KL bounds I_c.

i.

h. Variational Adversarial Imitation Learning(VAIL)

$$J(D, E) = \min_{D,E} \max_{\beta\geq 0} \mathbb{E}_{\mathbf{s}\sim\pi^*(\mathbf{s})} \left[\mathbb{E}_{\mathbf{z}\sim E(\mathbf{z}|\mathbf{s})} \left[-\log\left(D(\mathbf{z})\right)\right]\right] + \mathbb{E}_{\mathbf{s}\sim\pi(\mathbf{s})} \left[\mathbb{E}_{\mathbf{z}\sim E(\mathbf{z}|\mathbf{s})} \left[-\log\left(1 - D(\mathbf{z})\right)\right]\right]$$
$$+ \beta \left(\mathbb{E}_{\mathbf{s}\sim\tilde{\pi}(\mathbf{s})} \left[\mathrm{KL}\left[E(\mathbf{z}|\mathbf{s})||r(\mathbf{z})\right]\right] - I_c\right).$$

i.

ii. $\tilde{\pi} = \frac{1}{2}\pi^* + \frac{1}{2}\pi$ represents a mixture of the target policy and the agent's policy.

iii. Reward : $r_t = -\log\left(1 - D(\mu_E(\mathbf{s}))\right)$

i. Variational Adversarial Inverse Reinforcement Learning

i. 기본 GAIL form

1. $$D(\mathbf{s}, \mathbf{a}, \mathbf{s}') = \frac{\exp\left(f(\mathbf{s}, \mathbf{a}, \mathbf{s}')\right)}{\exp\left(f(\mathbf{s}, \mathbf{a}, \mathbf{s}')\right) + \pi(\mathbf{a}|\mathbf{s})},$$

2. $$f(\mathbf{s}, \mathbf{a}, \mathbf{s}') = g(\mathbf{s}, \mathbf{a}) + \gamma h(\mathbf{s}') - h(\mathbf{s}),$$

   a. G 와 H 는 학습된 Function

   b. $g(\mathbf{s})$ recovers the expert's true reward function $r^*(\mathbf{s})$

   c. : stochastic encoders $E_g(\mathbf{z}_g|\mathbf{s}), E_h(\mathbf{z}_h|\mathbf{s}),$

$$D(\mathbf{s},\mathbf{a},\mathbf{z}) = \frac{\exp\left(f(\mathbf{z}_g,\mathbf{z}_h,\mathbf{z}'_h)\right)}{\exp\left(f(\mathbf{z}_g,\mathbf{z}_h,\mathbf{z}'_h)\right)+\pi(\mathbf{a}|\mathbf{s})},$$

3.
4. 식 재정의
5. $\mathbf{z} = (\mathbf{z}_g, \mathbf{z}_h, \mathbf{z}'_h)$ and $f(\mathbf{z}_g, \mathbf{z}_h, \mathbf{z}'_h) = D_g(\mathbf{z}_g) + \gamma D_h(\mathbf{z}'_h) - D_h(\mathbf{z}_h)$.
6. 최종 Object Function

$$\begin{aligned} J(D,E) = \min_{D,E}\max_{\beta\geq 0} \quad & \mathbb{E}_{\mathbf{s},\mathbf{s}'\sim\pi^*(\mathbf{s},\mathbf{s}')}\left[\mathbb{E}_{\mathbf{z}\sim E(\mathbf{z}|\mathbf{s},\mathbf{s}')}\left[-\log\left(D(\mathbf{s},\mathbf{a},\mathbf{z})\right)\right]\right] \\ & + \mathbb{E}_{\mathbf{s},\mathbf{s}'\sim\pi(\mathbf{s},\mathbf{s}')}\left[\mathbb{E}_{\mathbf{z}\sim E(\mathbf{z}|\mathbf{s},\mathbf{s}')}\left[-\log\left(1-D(\mathbf{s},\mathbf{a},\mathbf{z})\right)\right]\right] \\ & + \beta\left(\mathbb{E}_{\mathbf{s},\mathbf{s}'\sim\tilde{\pi}(\mathbf{s},\mathbf{s}')}\left[\mathrm{KL}\left[E(\mathbf{z}|\mathbf{s},\mathbf{s}')||r(\mathbf{z})\right]\right] - I_c\right), \end{aligned}$$

   a.
7. where $\pi(s,s')$ denotes the joint distribution of successive states from a policy, and $E(\mathbf{z}|\mathbf{s},\mathbf{s}') = E_g(\mathbf{z}_g|\mathbf{s})\cdot E_h(\mathbf{z}_h|\mathbf{s})\cdot E_h(\mathbf{z}'_h|\mathbf{s}')$.

ii. GAIL vs VAIL 비교

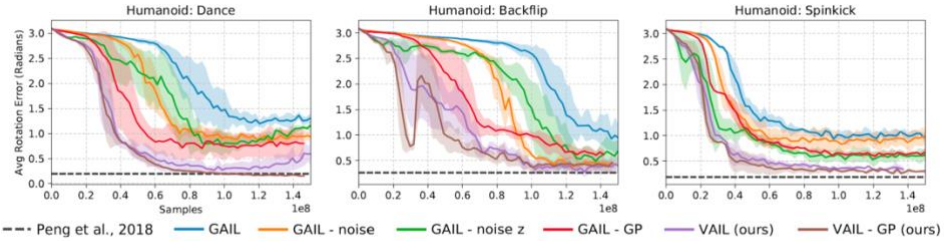


Figure 4: Learning curves comparing VAIL to other methods for motion imitation. Performance is measured using the average joint rotation error between the simulated character and the reference motion. Each method is evaluated with 3 random seeds.

Method	Backflip	Cartwheel	Dance	Run	Spinkick
BC	3.01	2.88	2.93	2.63	2.88
Merel et al., 2017	1.33 ± 0.03	1.47 ± 0.12	2.61 ± 0.30	0.52 ± 0.04	1.82 ± 0.35
GAIL	0.74 ± 0.15	0.84 ± 0.05	1.31 ± 0.16	0.17 ± 0.03	1.07 ± 0.03
GAIL - noise	0.42 ± 0.02	0.92 ± 0.07	0.96 ± 0.08	0.21 ± 0.05	0.95 ± 0.14
GAIL - noise z	0.67 ± 0.12	0.72 ± 0.04	1.14 ± 0.08	0.14 ± 0.03	0.64 ± 0.09
GAIL - GP	0.62 ± 0.09	0.69 ± 0.05	0.80 ± 0.32	0.12 ± 0.02	0.64 ± 0.04
VAIL (ours)	$\mathbf{0.36 \pm 0.13}$	0.40 ± 0.08	0.40 ± 0.21	0.13 ± 0.01	0.34 ± 0.05
VAIL - GP (ours)	0.46 ± 0.17	$\mathbf{0.31 \pm 0.02}$	$\mathbf{0.15 \pm 0.01}$	$\mathbf{0.10 \pm 0.01}$	$\mathbf{0.31 \pm 0.02}$
Peng et al., 2018	0.26	0.21	0.20	0.14	0.19

Table 1: Average joint rotation error (radians) on humanoid motion imitation tasks. VAIL outperforms the other methods for all skills evaluated, except for policies trained using the manually-designed reward function from (Peng et al., 2018).

iii.
iv. VAIL-GP(Gradient penalty)

$$\begin{aligned} J(D,E) = \min_{D,E} \quad & \mathbb{E}_{x\sim p^*(\mathbf{x})}\left[\mathbb{E}_{\mathbf{z}\sim E(\mathbf{z}|\mathbf{x})}\left[-\log\left(D(\mathbf{z})\right)\right]\right] + \mathbb{E}_{\mathbf{x}\sim G(\mathbf{x})}\left[\mathbb{E}_{\mathbf{z}\sim E(\mathbf{z}|\mathbf{x})}\left[-\log\left(1-D(\mathbf{z})\right)\right]\right] \\ & + w_{GP}\mathbb{E}_{x\sim p^*(\mathbf{x})}\left[\mathbb{E}_{\epsilon\sim\mathcal{N}(0,I)}\left[\frac{1}{2}||\nabla_x D(\mu_E(x)+\Sigma_E(x)\epsilon)||^2\right]\right] \\ \text{s.t.} \quad & \mathbb{E}_{\mathbf{x}\sim\tilde{p}(\mathbf{x})}\left[\mathrm{KL}\left[E(\mathbf{z}|\mathbf{x})||r(\mathbf{z})\right]\right] \leq I_c, \end{aligned}$$

1.

5. Experiments
   a. 3 개 실험 imitation learning, inverse reinforcement learning, image generation
      i. Imitation Learning

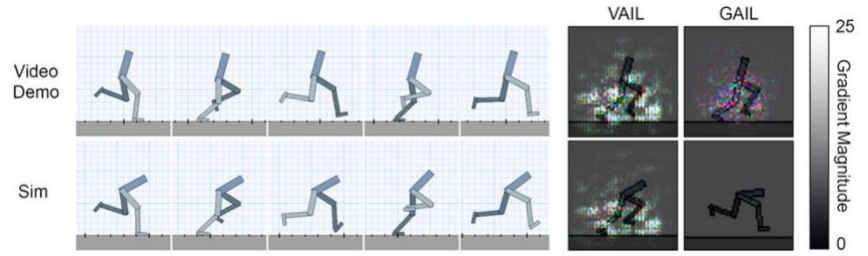
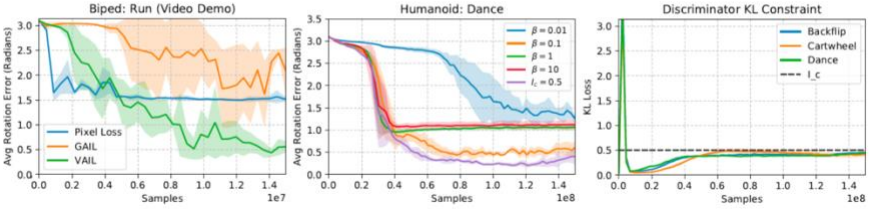
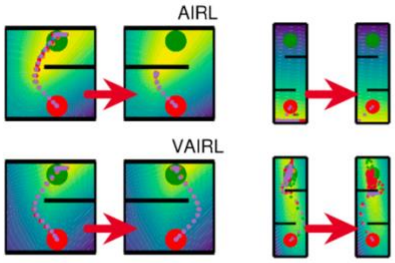


Figure 5: **Left:** Snapshots of the video demonstration and the simulated character trained with VAIL. The policy learns to run by directly imitating the video. **Right:** Saliency maps that visualize the magnitude of the discriminator's gradient with respect to all channels of the RGB input images from both the demonstration and the simulation. Pixel values are normalized between $[0, 1]$.



1.

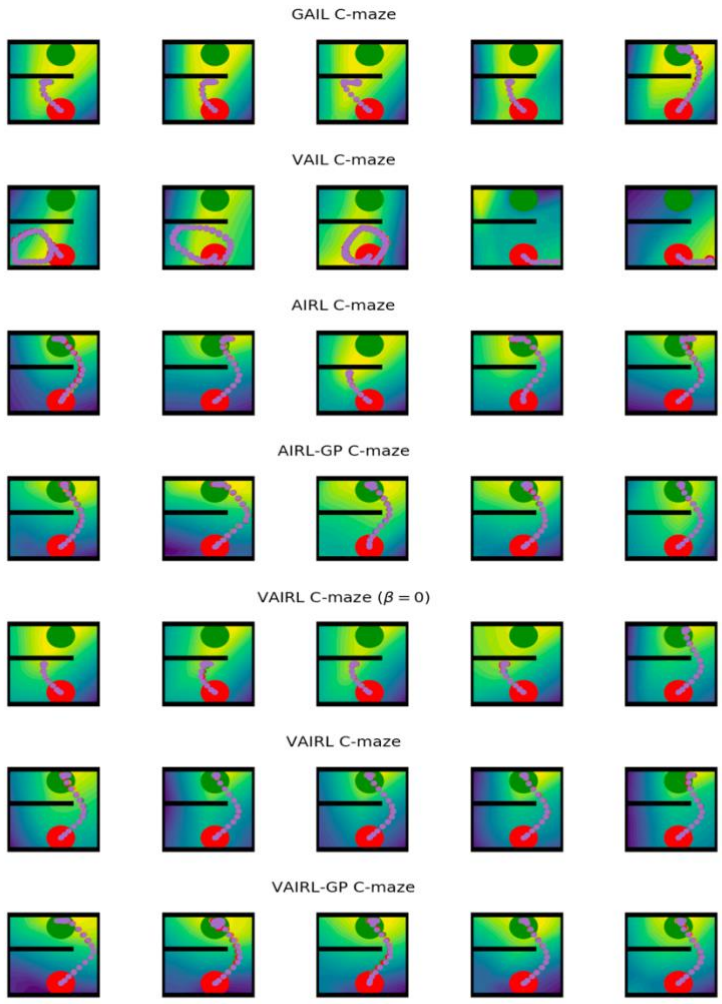
ii. Inverse Reinforcement Learning



Method	Transfer environments	
	C-maze	S-maze
GAIL	-24.6±7.2	1.0±1.3
VAIL	-65.6±18.9	20.8±39.7
AIRL	-15.3±7.8	-0.2±0.1
AIRL - GP	**-9.14±0.4**	-0.14±0.3
VAIRL ($\beta = 0$)	-25.5±7.2	62.3±33.2
VAIRL (ours)	-10.0±2.2	74.0±38.7
VAIRL - GP (ours)	-9.18±0.4	**156.5±5.6**
TRPO expert	-5.1	153.2

Figure 7: **Left:** C-Maze and S-Maze. When trained on the training maze on the left, AIRL learns a reward that overfits to the training task, and which cannot be transferred to the mirrored maze on the right. In contrast, VAIRL learns a smoother reward function that enables more-reliable transfer. **Right:** Performance on flipped test versions of our two training mazes. We report mean return (± std. dev.) over five runs, and the mean return for the expert used to generate demonstrations.

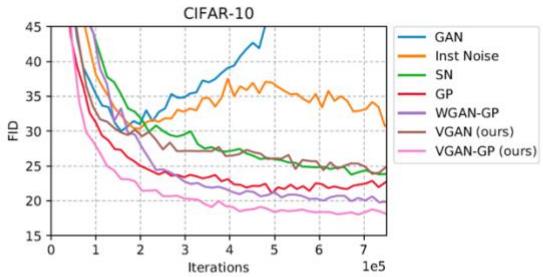
1.



'igure 14: Visualizations of recovered reward functions transferred to the mirrored C-maze. Also hown are trajectories executed by policies trained to maximize the corresponding reward in the new nvironment.

2.

iii. image generation



Method	FID
GAN	63.6
Inst Noise	30.7
SN	23.9
GP	22.6
WGAN-GP	19.9
VGAN (ours)	24.8
VGAN-SN (ours)	71.8
VGAN-GP (ours)	**18.1**

Figure 8: Comparison of VGAN and other methods on CIFAR-10, with performance evaluated using the Fréchet Inception Distance (FID).

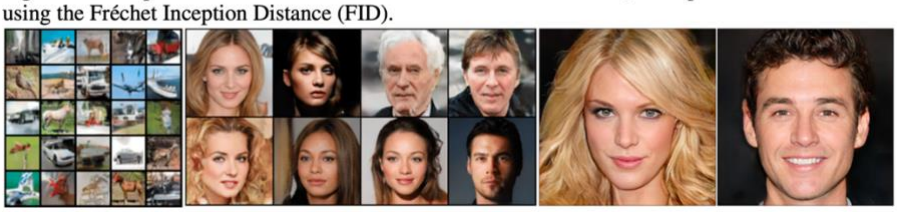


Figure 9: VGAN samples on CIFAR-10, CelebA 128×128, and CelebAHQ 1024×1024.

1.
2. FDI(Fre'chet Inception Distance)
    a. Real Data 와 생성된 데이터 간에 Feature Space 상 거리

Figure 18: VGAN samples on CelebA HQ (Karras et al., 2018) 1024 × 1024 resolution at 300k iterations. Models are trained from scratch at full resolution, without the progressive scheme proposed by Karras et al. (2017).

3.

6. Reference

    a. https://curt-park.github.io/2019-05-05/vdb/

- 용어:
    - Mutual Information(상호 정보량):
        - 두 확률 변수가 얼마나 관련있는지를 계량화
        - 전혀 관계가 없다면 0, 완전 관련이라면 1