LSGAN & ACGAN

GAN: N-STEP FURTHER

REMARK

- Binary Cross-Entropy Loss

$$BCE(x) = -\frac{1}{N}\sum_{i=1}^{N} y_i \log\left(h(x_i;\theta)\right) + (1-y_i)\log\left(1-h(x_i;\theta)\right)$$

  - 확률적 분류 모델의 학습
  - 입력에 대한 확률적 예측이 ground-truth 확률과 최대한 유사하도록

REMARK: DCGAN

```
# Discriminator
inputs = Input(shape=img_shape, name='discriminator_input')
discriminator = build_discriminator(inputs)
optimizer = RMSprop(lr=lr, decay=decay)
discriminator.compile(loss='binary_crossentropy', optimizer=optimizer, metrics=['accuracy'])
discriminator.summary()

# Adversarial
discriminator.trainable = False  # Fix weights  # Boolean flag at compiling

inputs = Input(shape=z_shape, name='generator_input')
adversarial = Model(inputs, discriminator(generator(inputs)), name=model_name)
optimizer = RMSprop(lr=lr * 0.5, decay=decay * 0.5)
adversarial.compile(loss='binary_crossentropy', optimizer=optimizer, metrics=['accuracy'])
adversarial.summary()
```

LSGAN

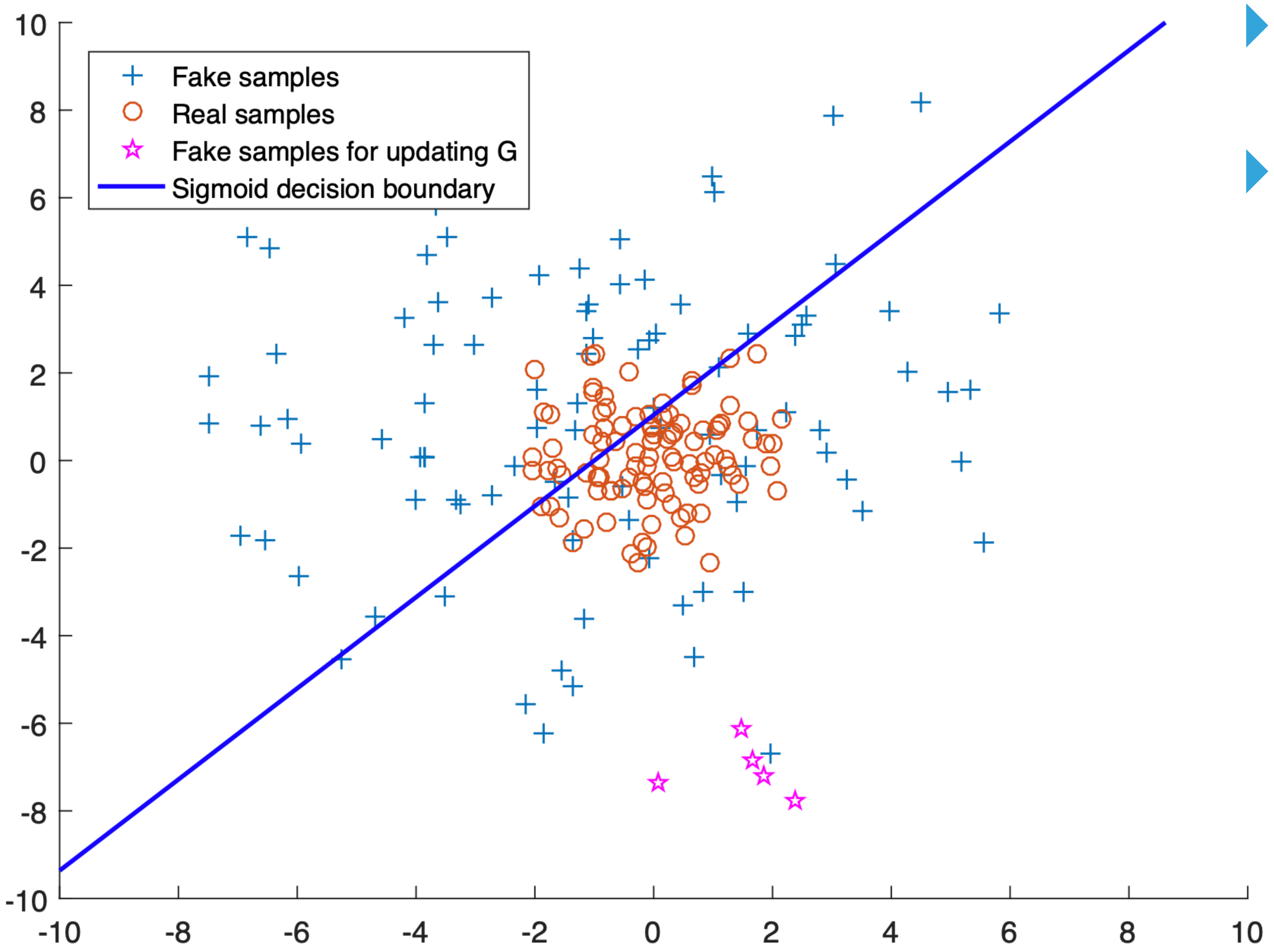
LSGAN

- Least Squares Generative Adversarial Networks
- Ref: Mao, Xudong, et al. "Least squares generative adversarial networks." Proceedings of the IEEE International Conference on Computer Vision. 2017.

GAN

- 기존 GAN에서의 sigmoid cross entropy loss (=binary cross entropy loss)
  - Gradient vanishing 문제를 야기함

GAN

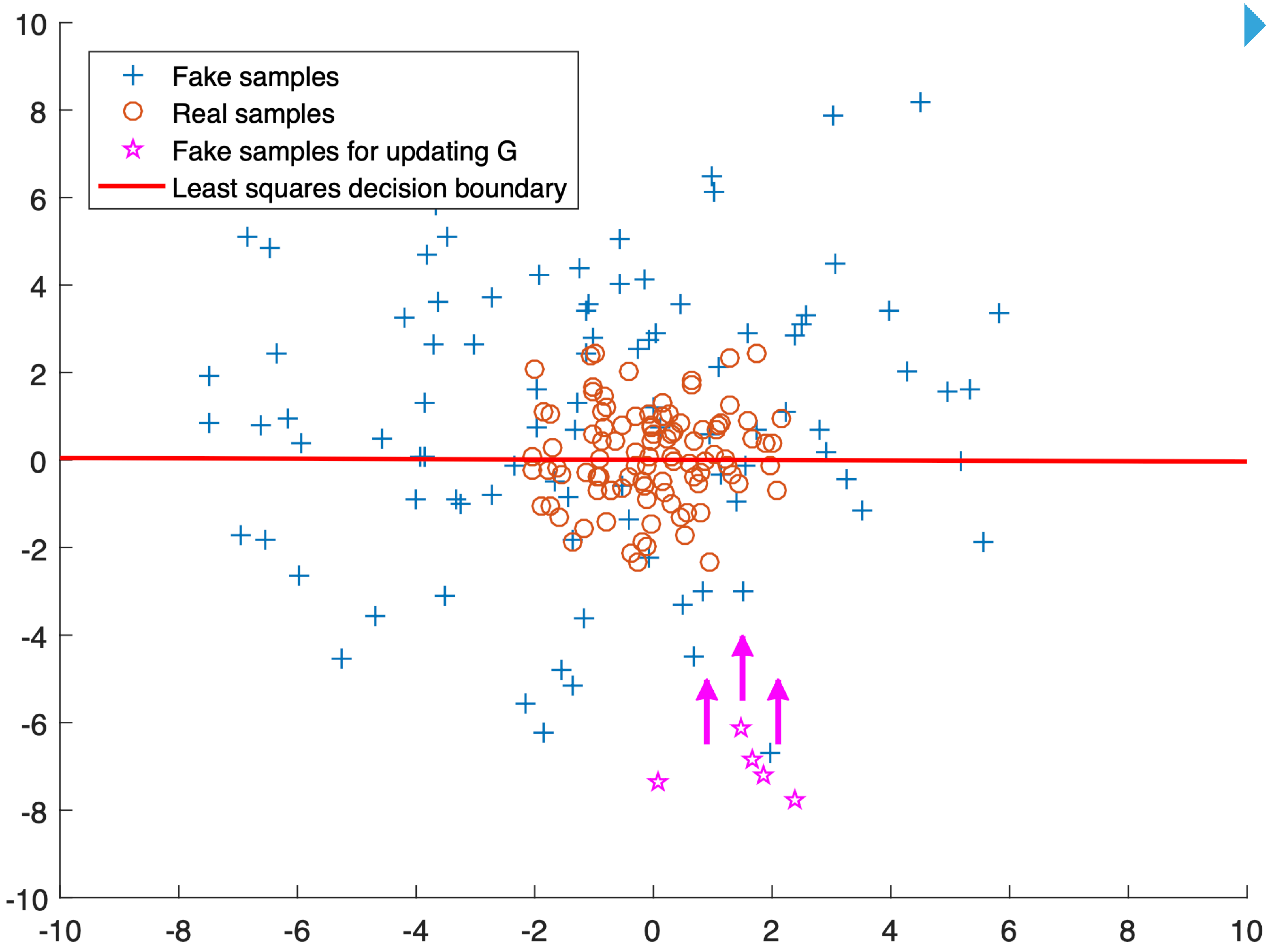


- 결정 경계 아래: 진짜로 구별
- 별(마젠타): 가짜이지만 진짜로 판단 중
  - 생성기 업데이트에 도움을 주지 않음
  - Gradient vanishing

LSGAN

- LSGAN의 least square
  - 가짜 데이터를 decision boundary에 가깝게 이동
  - 문제를 해결할 수 있음

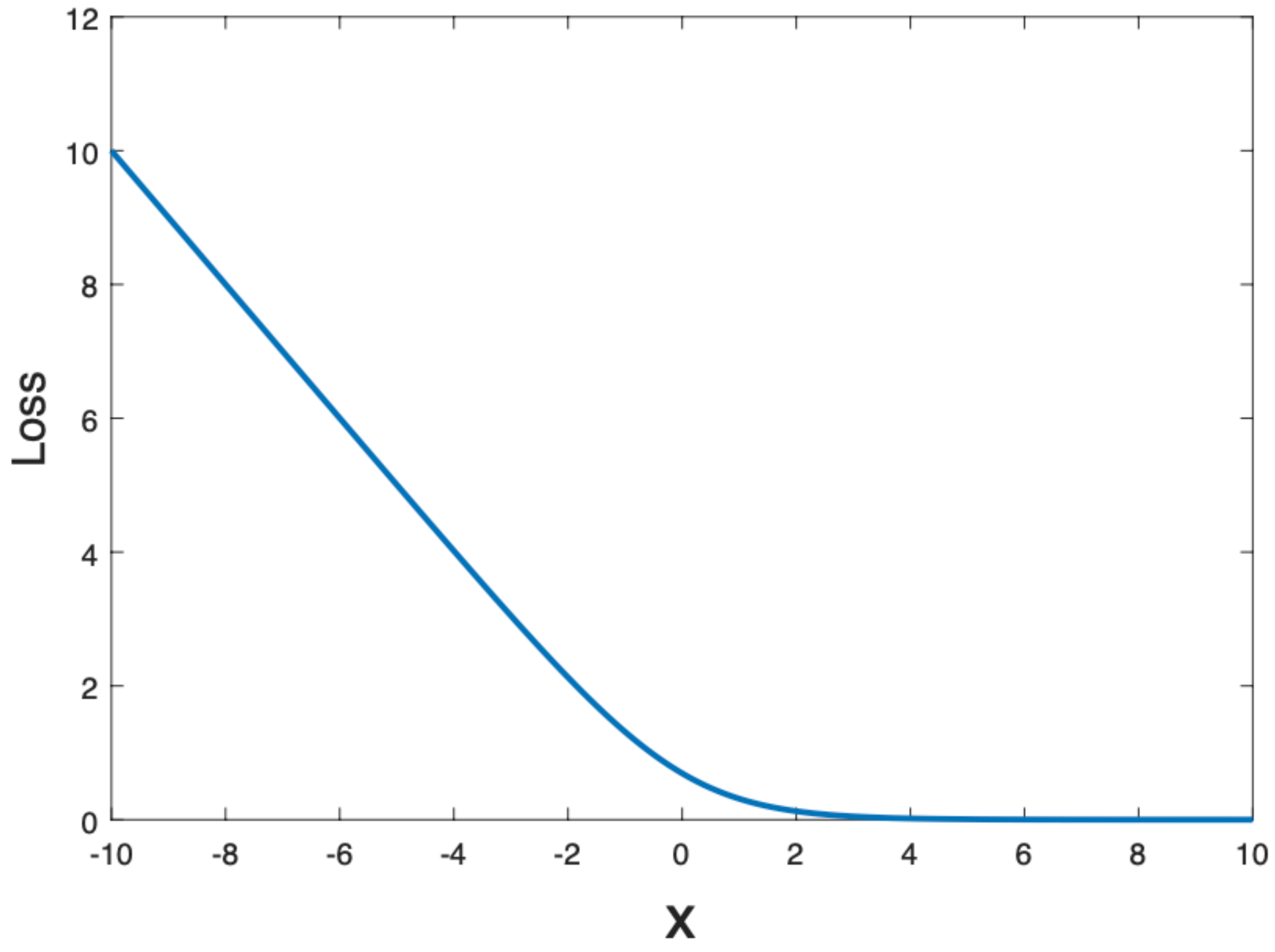
LSGAN

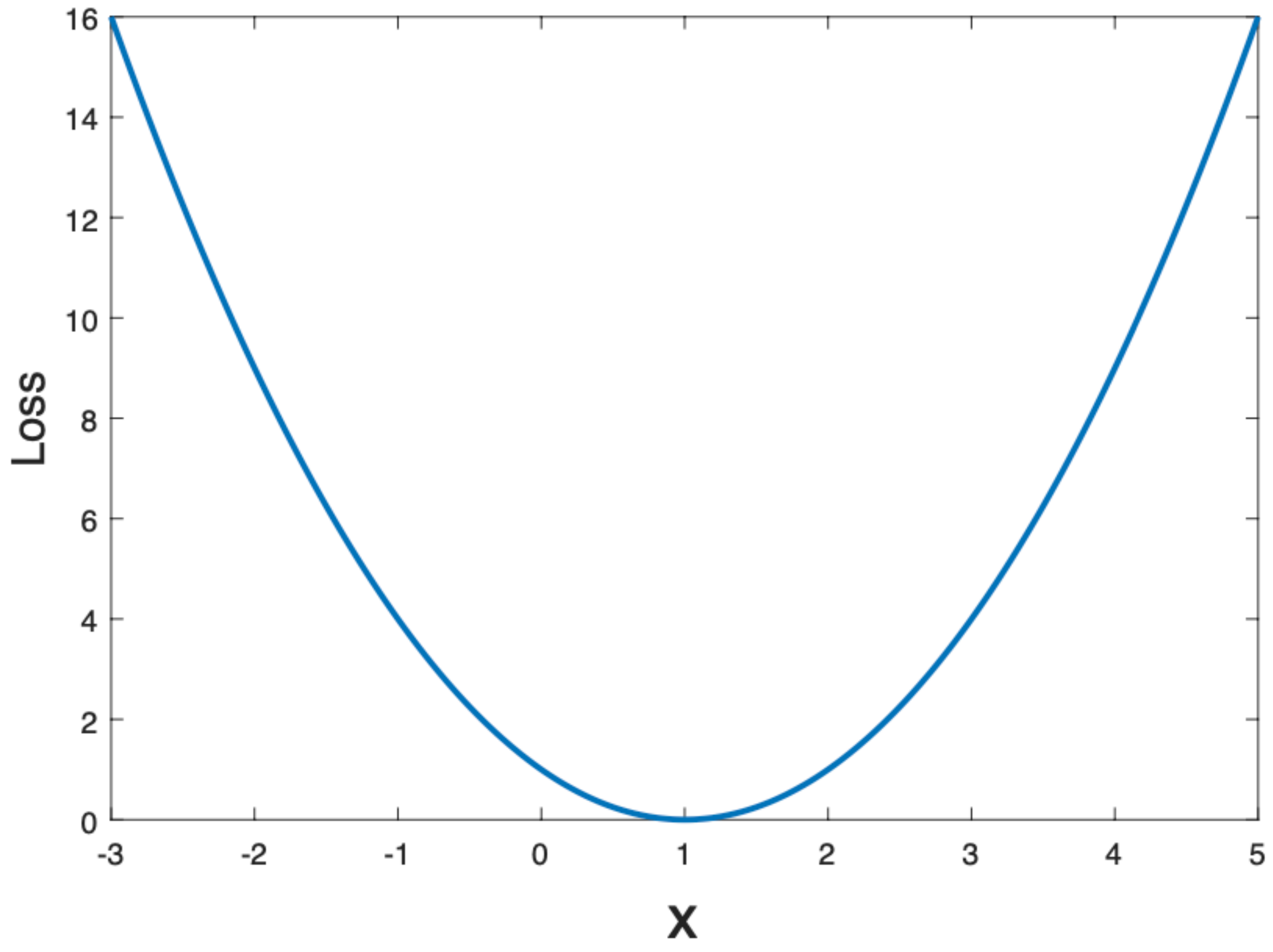


- 결정 경계에서 멀리 떨어져 있을 수록 패널티
  - 별(마젠타) 데이터를 진짜 데이터 근처로
  - Least square loss 사용

BENEFITS

- Sigmoid cross entropy loss function (vs.) Least squares loss function





LSGAN

- 기존의 GAN:
  - $$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[logD(x)] + \mathbb{E}_{z\sim p_z(z)}[log(1-D(G(z)))]$$

- LSGAN
  - $$\min_D V_{LSGAN}(D) = \frac{1}{2}\mathbb{E}_{x\sim p_{data}(x)}[(D(x)-b)^2] + \frac{1}{2}\mathbb{E}_{z\sim p_z(z)}[(D(G(z))-a)^2]$$
    $$\min_G V_{LSGAN}(G) = \frac{1}{2}\mathbb{E}_{z\sim p_z(z)}[(D(G(z))-c)^2]$$

LSGAN

- LSGAN
  - $$\min_D V_{LSGAN}(D) = \frac{1}{2}\mathbb{E}_{x\sim p_{data}(x)}[(D(x)-b)^2] + \frac{1}{2}\mathbb{E}_{z\sim p_x(z)}[(D(G(z))-a)^2]$$
    $$\min_G V_{LSGAN}(G) = \frac{1}{2}\mathbb{E}_{z\sim p_z(z)}[(D(G(z))-c)^2]$$
  - a: fake label
    b: real label
    c: G 입장에서 D가 진짜라 믿게 하고픈 fakes

LSGAN

- LSGAN
  - $$\min_D V_{LSGAN}(D) = \frac{1}{2}\mathbb{E}_{x\sim p_{data}(x)}\left[\boxed{(D(x)-b)^2}\right] + \frac{1}{2}\mathbb{E}_{z\sim p_z(z)}\left[\boxed{(D(G(z))-a)^2}\right]$$
    $$\min_G V_{LSGAN}(G) = \frac{1}{2}\mathbb{E}_{z\sim p_z(z)}\left[(D(G(z))-c)^2\right]$$

- 최소가 되기 위해:
  - $D(x) \rightarrow reals$
  - $D(G(z)) \rightarrow fakes$

LSGAN

- LSGAN
  - $$\min_D V_{LSGAN}(D) = \frac{1}{2}\mathbb{E}_{x \sim p_{data}(x)}[(D(x) - b)^2] + \frac{1}{2}\mathbb{E}_{z \sim p_x(z)}[(D(G(z)) - a)^2]$$
    $$\min_G V_{LSGAN}(G) = \frac{1}{2}\mathbb{E}_{z \sim p_z(z)}[(\boxed{D(G(z)) - c)^2}]$$

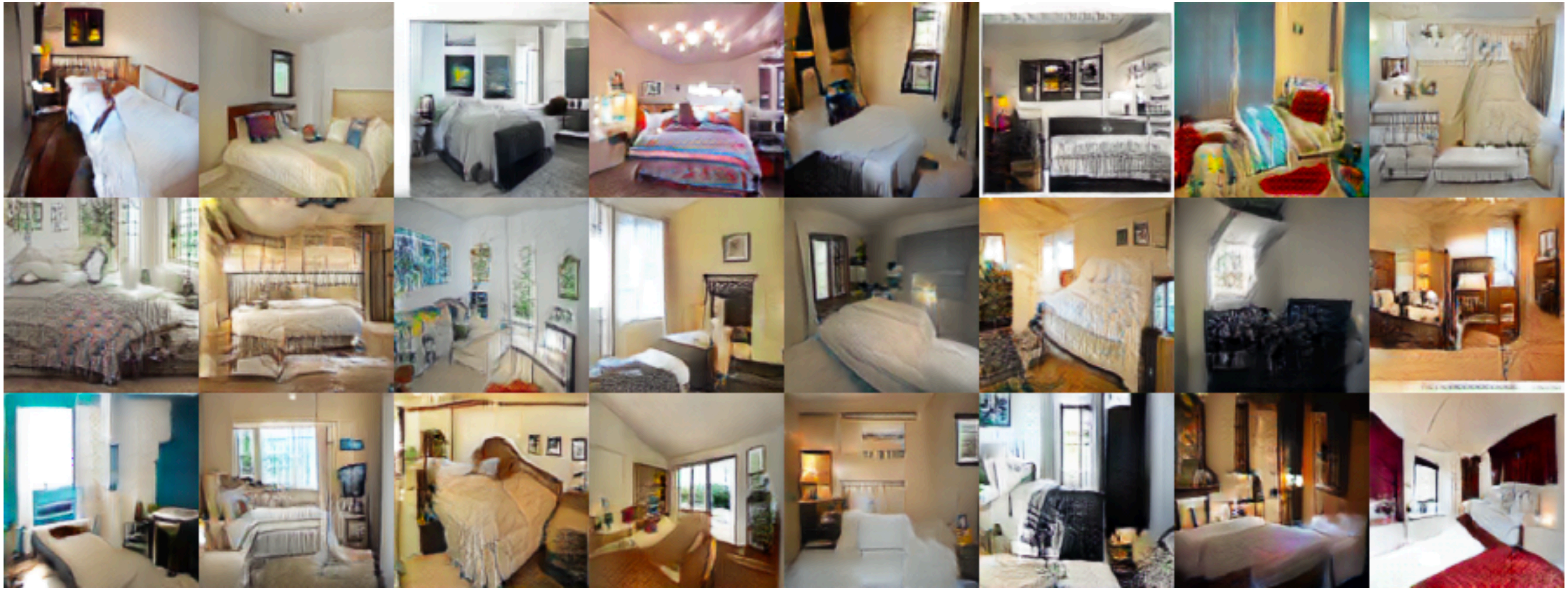
- 최소가 되기 위해:
  - $D(G(z)) \rightarrow fakes\ imitating\ reals$

EXPERIMENTS

- DCGANs:

EXPERIMENTS

- LSGANs:



EXPERIMENTS

- 안정성
  - Mode collapse의 유무 (NO가 좋음)

Optimizer	BN_G Adam	BN_G RMSProp	BN_{GD} Adam	BN_{GD} RMSProp
Regular GANs	YES	NO	YES	YES
LSGANs	**NO**	NO	YES	**NO**

  - BN: Batch Normalization (G, D) 제외
  - Adam (vs.) RMSProp

ACGAN

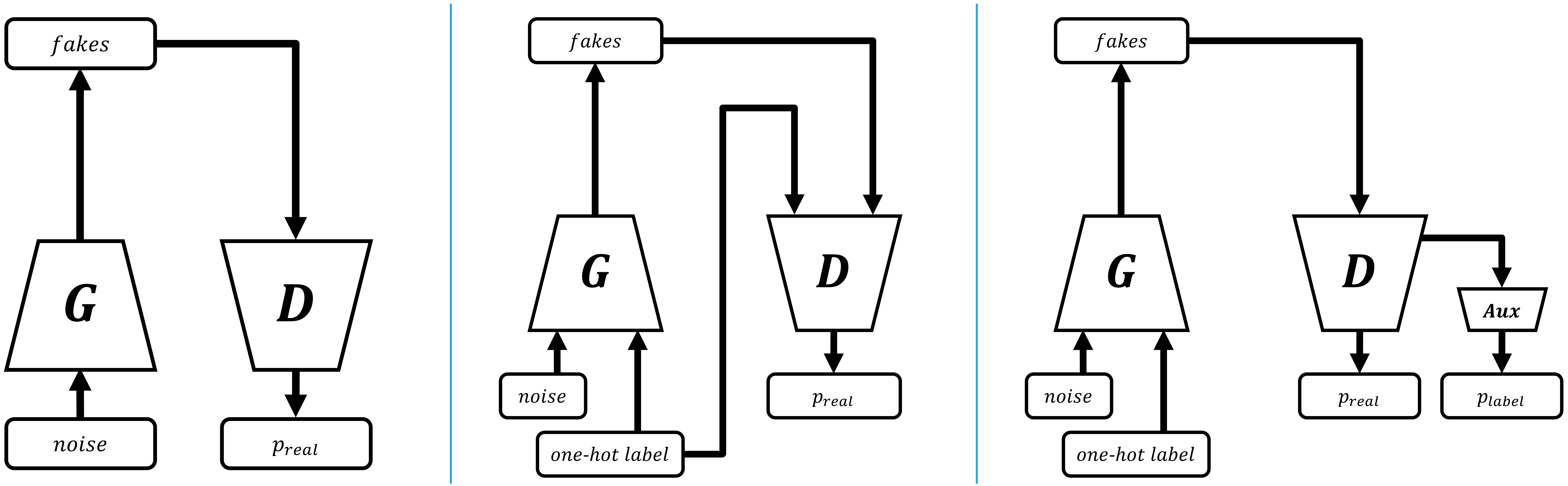
ACGAN

- Ref: Odena, Augustus, Christopher Olah, and Jonathon Shlens. "Conditional image synthesis with auxiliary classifier gans." Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org, 2017.

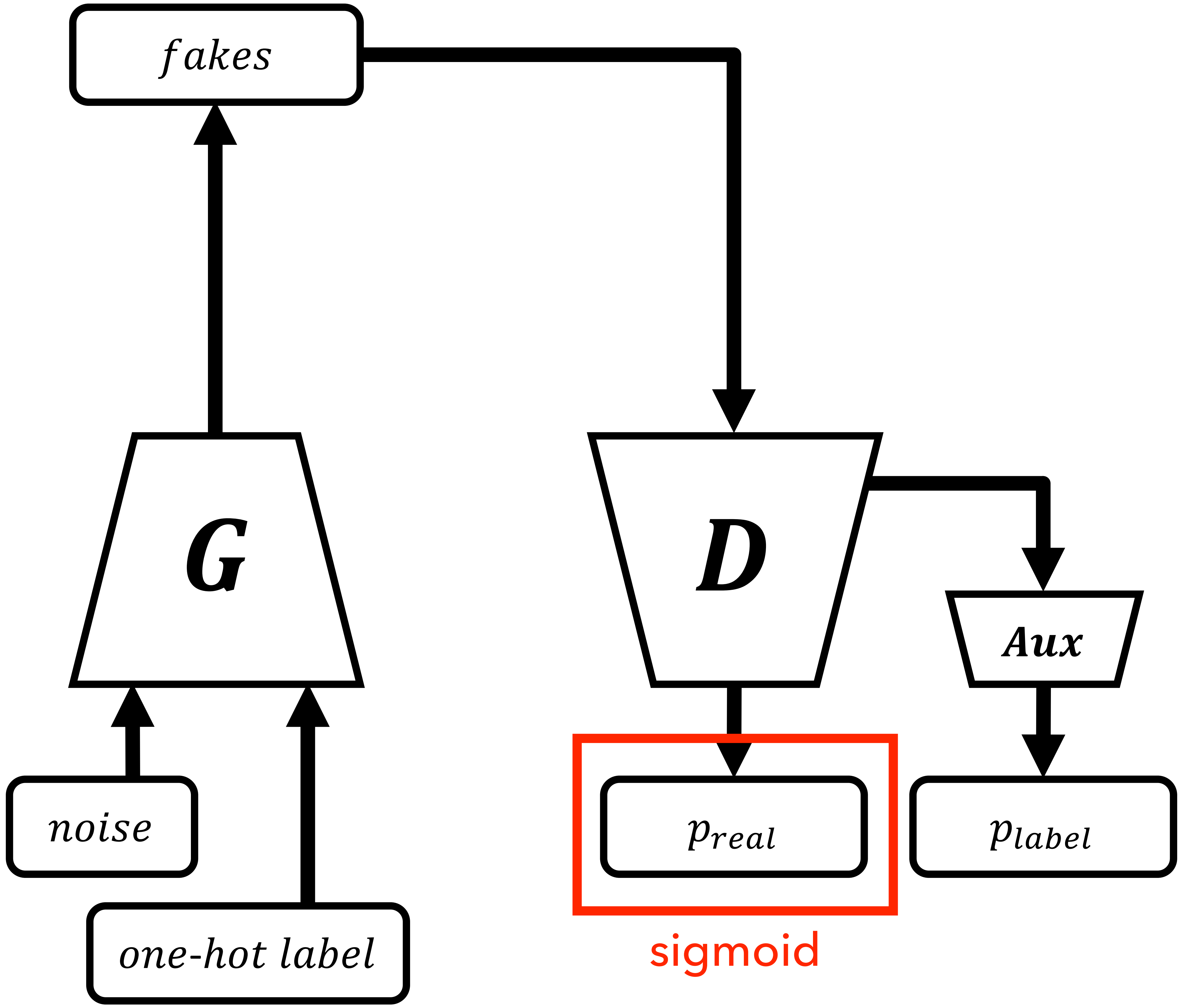
ACGAN

- 보조 분류기(Auxiliary Classifier) GANs
- CGAN 판별기
  - 입력: 이미지와 레이블
  - 출력: 이미지가 진짜일 확률
- ACGAN 판별기
  - 입력: 이미지
  - 출력: 이미지가 진짜이면서 클래스 레이블일 확률

DCGAN VS CGAN VS ACGAN

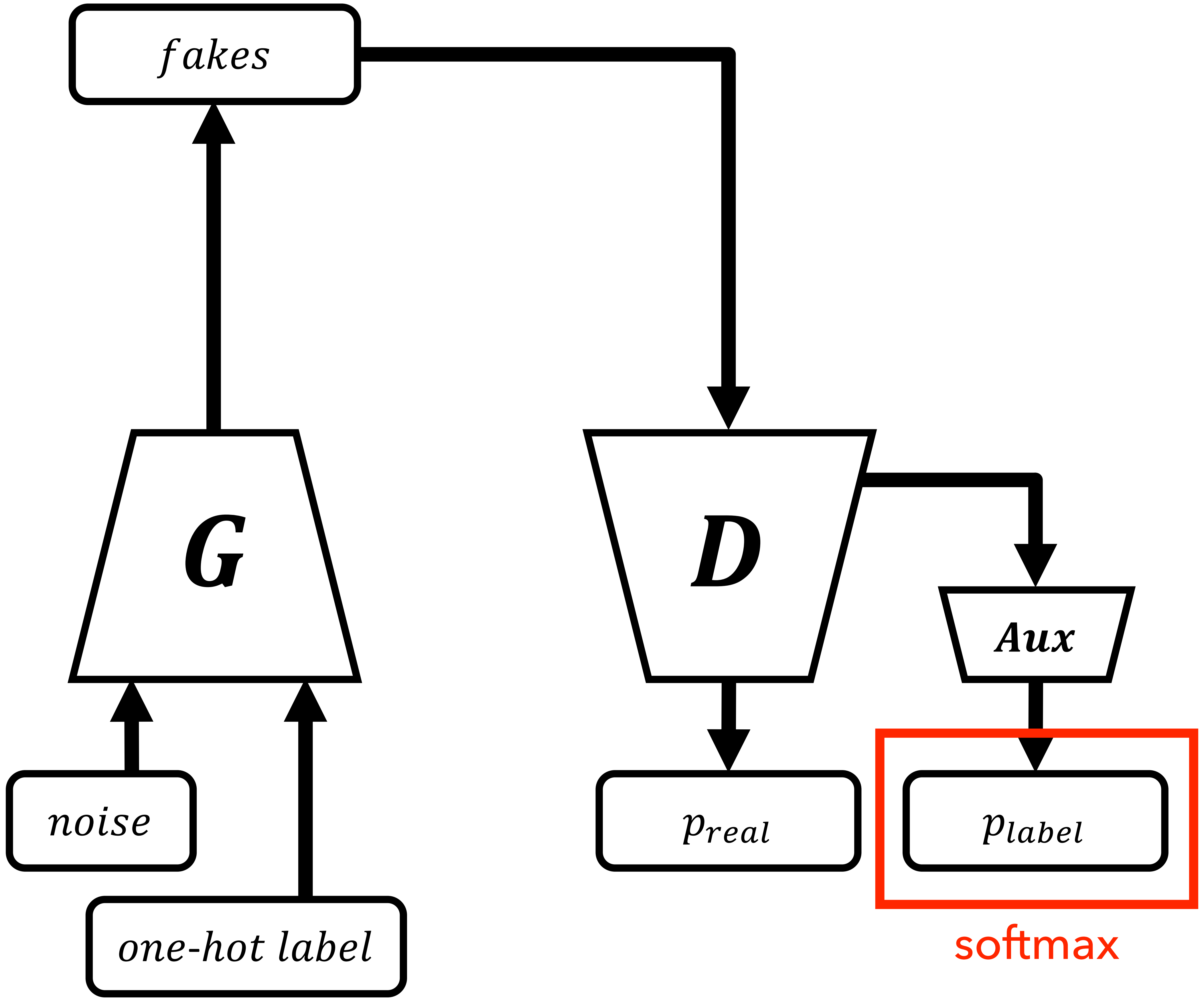


ACGAN



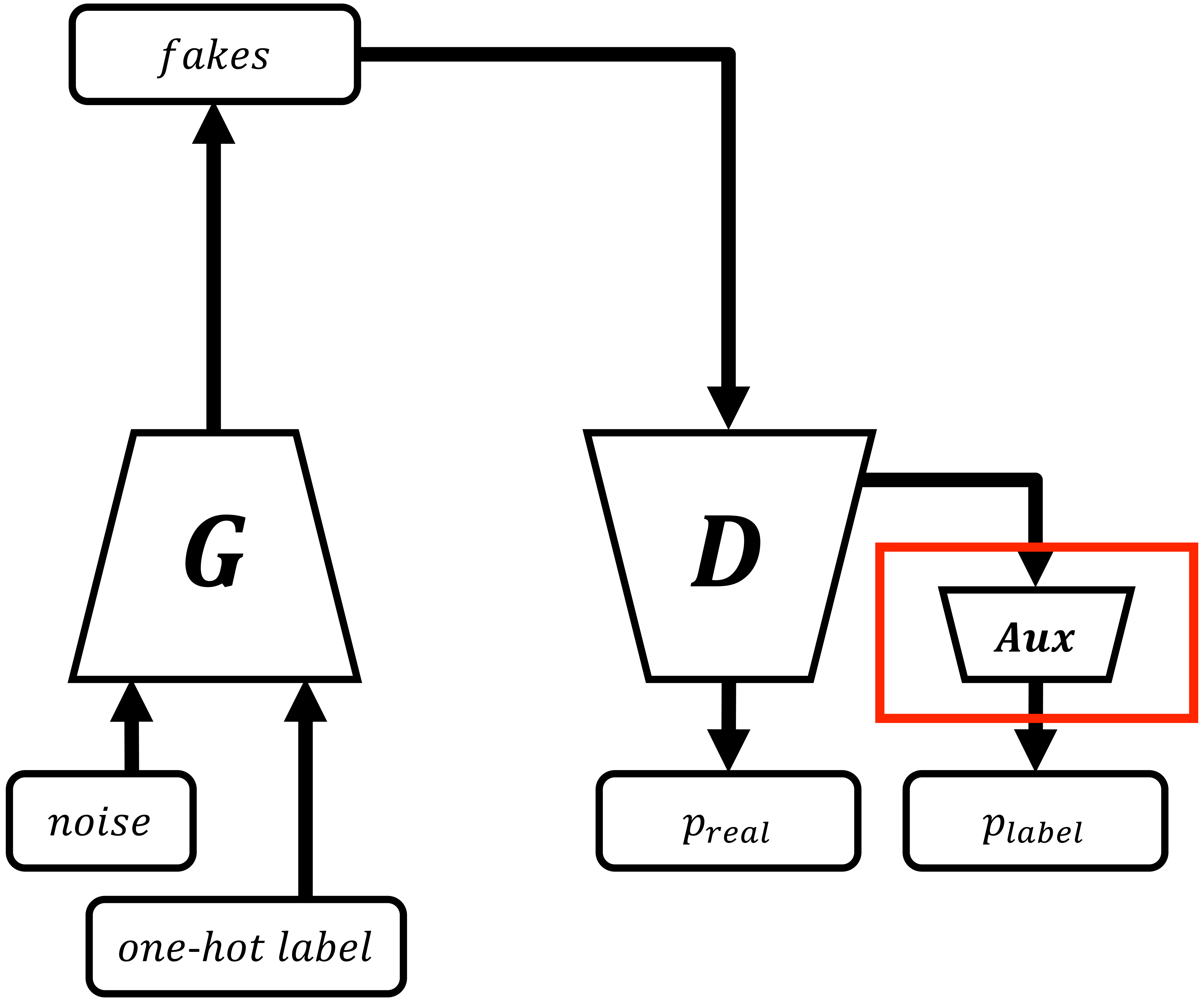
- p_{real} : Binary Cross-Entropy Loss

ACGAN



- p_{label} : Categorical Cross-Entropy Loss

ACGAN



- Aux : Auxiliary class decoder network

AUX

- 보조 클래스 디코더 네트워크(Auxiliary class decoder network)
  - 부가 정보인 레이블 정보를 재구성
  - 네트워크에 부가 작업을 시킴으로써 본래 작업의 성능을 향상
    - 본래 작업인 가짜 이미지 생성

ACGAN

- 손실 함수
- $L_S = E[log\, p(S = real | X_{real})] + E[log\, p(S = fake | X_{fake})]$
- $L_C = E[log\, p(C = c \quad | X_{real})] + E[log\, p(C = c \quad | X_{fake})]$
  - S는 Source (data)
  - C는 Class (label)

ACGAN

- $L_S = E[log\, p(S = real|X_{real})] + E[log\, p(S = fake|X_{fake})]$
- $L_C = E[log\, p(C = c \quad |X_{real})] + E[log\, p(C = c \quad |X_{fake})]$
- 판별기
  - $L_S + L_C$ 를 최대화
- 생성기
  - $-L_S + L_C$ 를 최대화

ACGAN

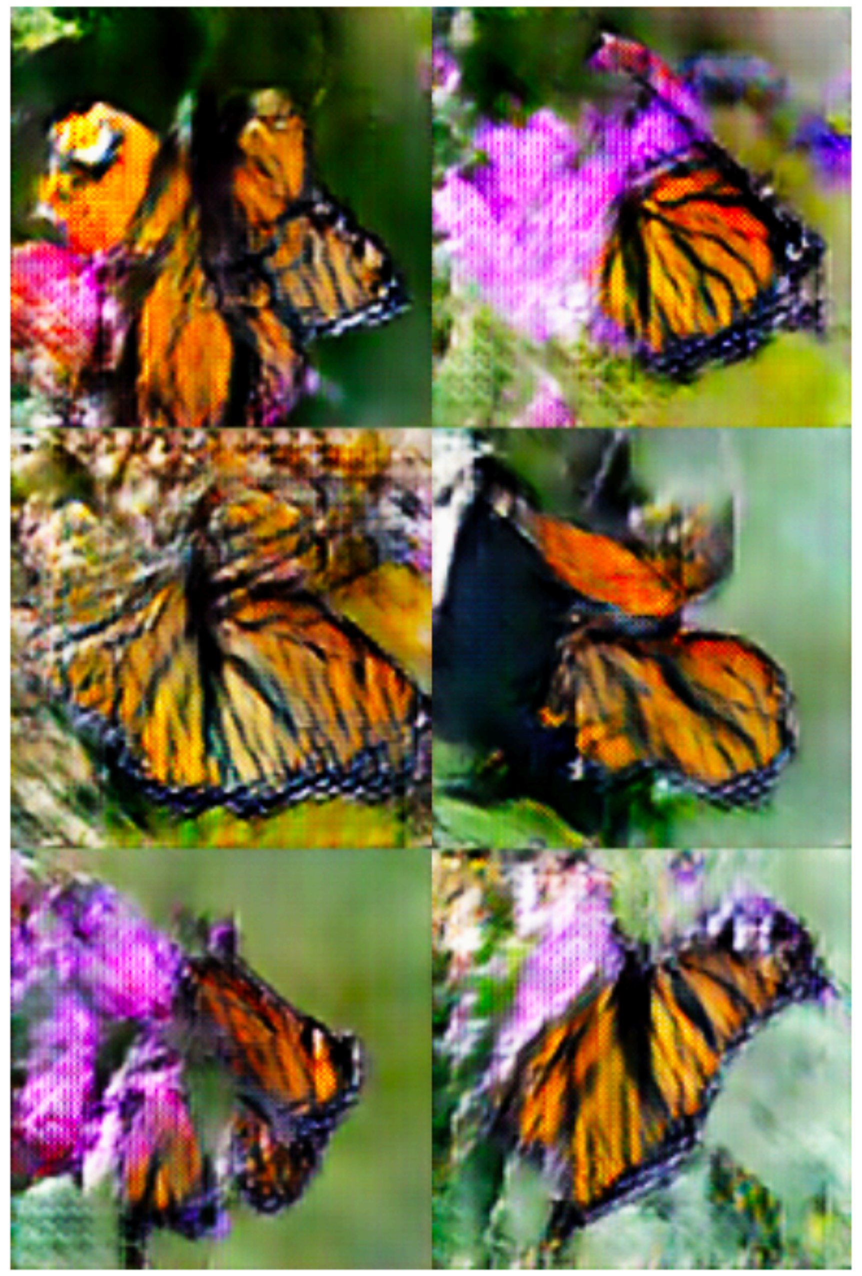
- 판별기
  - $L_S + L_C$ 를 최대화
  - $-L_S - L_C$ 를 최소화
  - $$L^{(D)} = -\mathbb{E}_{x \sim P_{data}} \log D(x) - \mathbb{E}_z \log(1 - D(G(z \mid y))) \\ -\mathbb{E}_{x \sim P_{data}} \log p(c \mid x) - \mathbb{E}_z \log p(c \mid G(z \mid y))$$

ACGAN

- 생성기
  - $-L_S + L_C$ 를 최대화
  - $L_S - L_C$ 를 최소화
  - $L^{(G)} = -\boxed{\mathbb{E}_z \log D(G(z|y))}$
    $-\mathbb{E}_z \log p(c|G(z|y))$
    - 가짜를 진짜라 말해야 함: L_S 의 반대 = $-L_S$
    - Real data 상관 없음

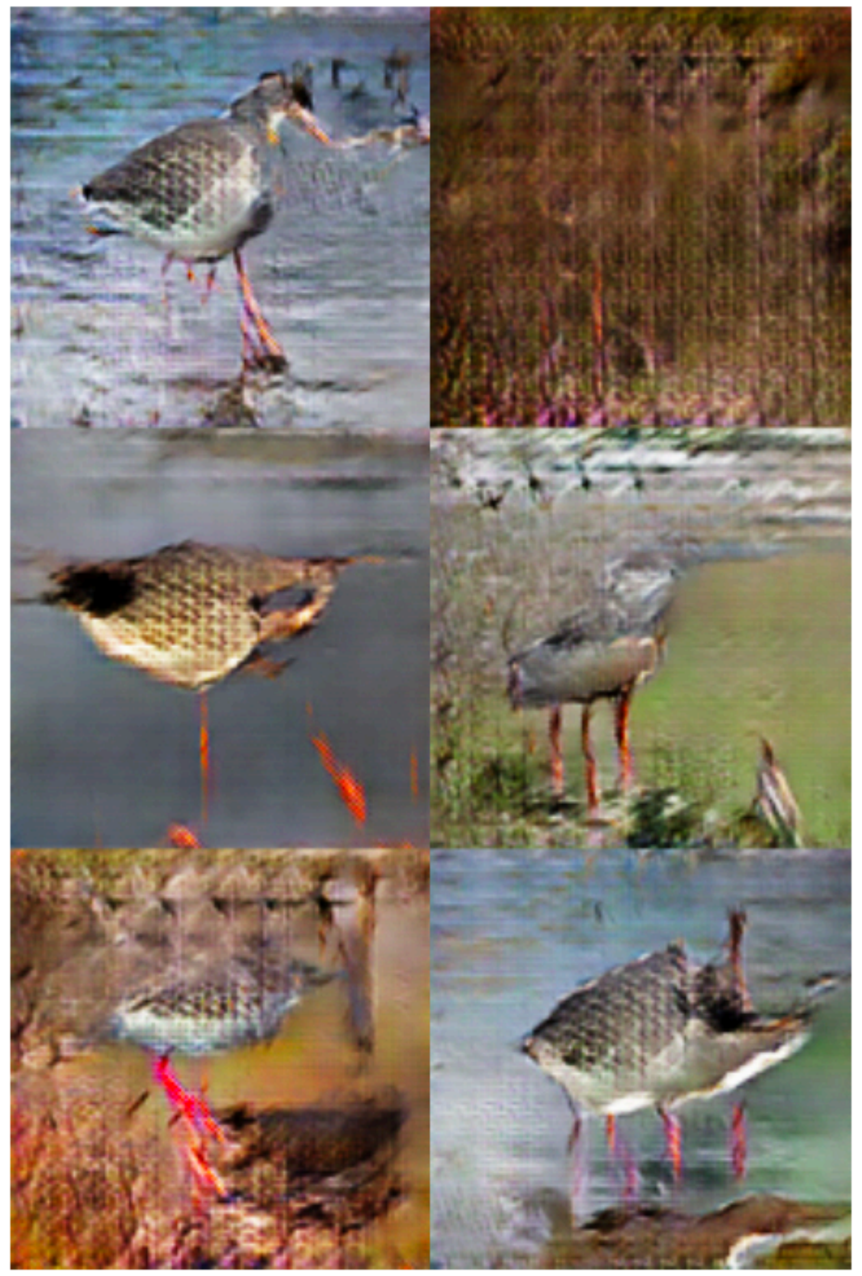
5 CLASSES ON THE IMAGENET DATASET

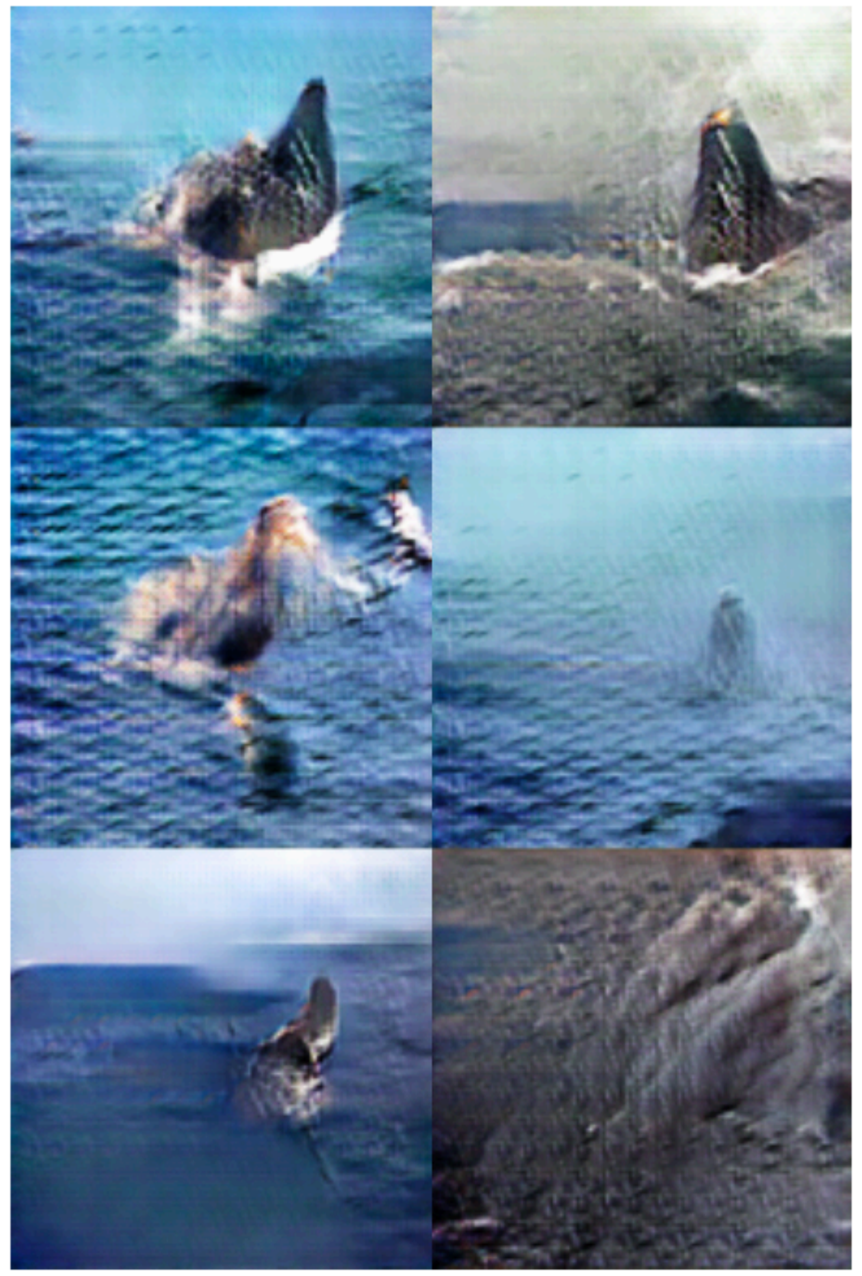
- 128×128 해상도의 가짜 이미지들
  - 당시(2016) 기준으로 고해상도


monarch butterfly


goldfinch


daisy


redshank


grey whale

LSGAN & ACGAN

GAN: N-STEP FURTHER