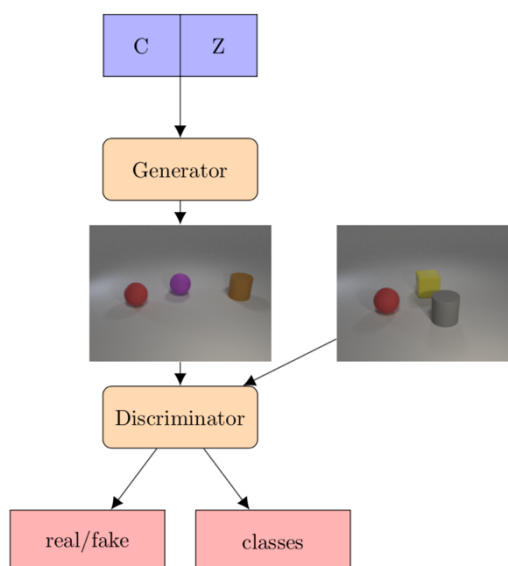
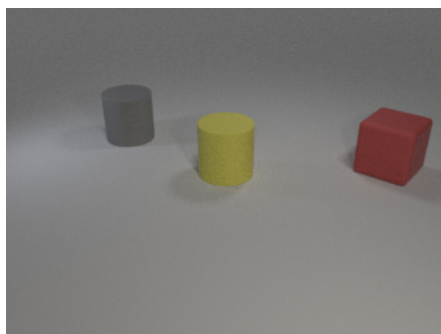


## 1. Introduction

在這次的 lab 中要實作一個 conditional GAN，希望使用不同的 condition 來產生該類別的圖片。cGAN 架構如下圖所示。



這次使用的 dataset 共有 24 類，由 3 種形狀和 8 種顏色所構成，形狀為 cube、sphere、cylinder，而顏色為 gray、red、blue、green、brown、purple、cyan、yellow，而每一張圖中有 1~3 個不等的物體。下圖為 dataset 中的一張圖片。



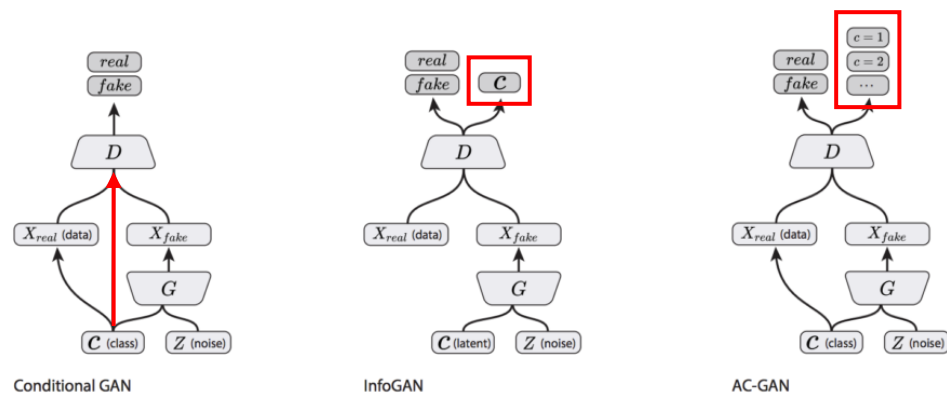
## 2. Implementation details

A. Describe how you implement your model, including your choice of cGAN, model architectures, and loss functions.

### ■ cGAN

cGAN 的部分我使用的是 AC-GAN (Auxiliary Classifier GAN)，可以將 AC-GAN 看成是 cGAN+infoGAN，架構比較圖如下圖所示。

和一般的 cGAN 不同的是，AC-GAN 不直接將  $c$  輸入 discriminator，而是像 infoGAN 一樣，讓 discriminator 多一個分類器當輔助，來預測 condition。



### ■ model architectures

#### ○ DCGAN: 結合 CNN 與 GAN

- generator 中使用 transposed convolutional layer
- 刪除 pooling layer

#### ○ Generator

- 將 latent vector  $z$  與 condition  $c$  接在一起，輸入 generator
- 使用 ConvTranspose2d、BatchNorm2d、ReLU，重複四層
- 最後一層用 ConvTranspose2d 與 Tanh

#### ○ Discriminator

- 將 latent vector  $z$  與 condition  $c$  接在一起，輸入 discriminator
- 使用 Linear、LeakyReLU，重複三層
- 使用 Linear、Sigmoid，重複兩層

## ■ loss functions

AC-GAN 有兩種 loss : Adversarial loss 與 Classification loss 。

下列為 generator 的 adversarial loss, classification loss 。

$$L_{adv}(G) = \mathbb{E}_{z \sim p_z, c \sim p_c} [\log(1 - D(G(z, c)))]$$

$$L_{cls}(G) = \mathbb{E}_{z \sim p_z, c \sim p_c} [L_D(c|G(z, c))]$$

下列為 discriminator 的 adversarial loss, classification loss 。

$$L_{adv}(D) = -\mathbb{E}_{x \sim p_{data}} [\log D(x)] - \mathbb{E}_{z \sim p_z, c \sim p_c} [\log(1 - D(G(z, c)))]$$

$$L_{cls}(D) = \mathbb{E}_{x \sim p_{data}} [L_D(c_x|x)]$$

為了調節 adversarial loss 與 classification loss 所佔的比例，加入了 weight，在我做的實驗中  $\text{weight} \in [0, 1]$  。

Loss function = Adversarial loss + weight \* Classification loss

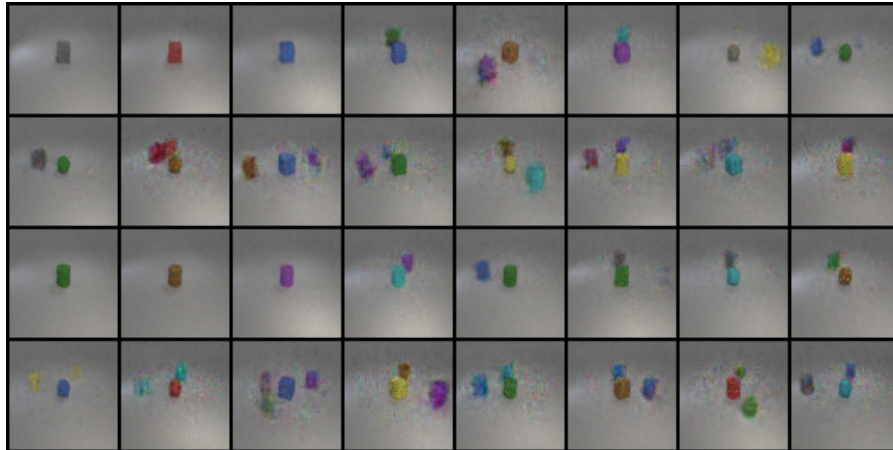
## B. Specify the hyperparameters

- Learning rate: 0.0002
- Epoch: 1000
- Batch size: 64
- Latent size: 40
- Weight of classification loss: 0.5

### 3. Results and discussion

A. Show your results based on the testing data.

Score: 0.7916666666666666



B. Discuss the results of different models architectures.

- Discriminator:

原本在 discriminator 中使用 convolutional layer，不過準確率一直無法提升，且 generator 的 adversarial loss 比其他 loss 高出很多，代表 discriminator 太強、generator 太弱，因此我將 discriminator 中的 convolutional layer 改成 fully connected layer，這樣的架構能讓 generator 與 discriminator 的能力較為相近，使得 generator 的 adversarial loss 變小許多，且能得到較高的準確率。

- Loss weight

Loss function = Adversarial loss + weight \* Classification loss  
Adversarial loss 越低表示 generator 產生的照片越好，而 Classification loss 越低表示產生的圖片與 condition 的相關性越高。因此在訓練 model 時要權衡兩種 loss 的比重。一開始設 weight=1，也就是 Adversarial loss 和 Classification loss 一樣重要，發現 classification loss 很快就會變成 0，不過 discriminator 的

adversarial loss 一直無法下降，並且產生出來的圖片雖然 condition ( 形狀、顏色 ) 是相近的，但圖片的品質是不好的。

因此我試著降低 weight 的數值，雖然 classification loss 下降得比較慢，但隨著 epoch 的數量上升，產生的圖片還是能夠學到 condition 的形狀、顏色，並且圖片也有較好的品質，因此 adversarial loss 在這個實驗中更為重要。