

# **Generative adversarial networks (GAN)**

Котенко А. ФВЕ

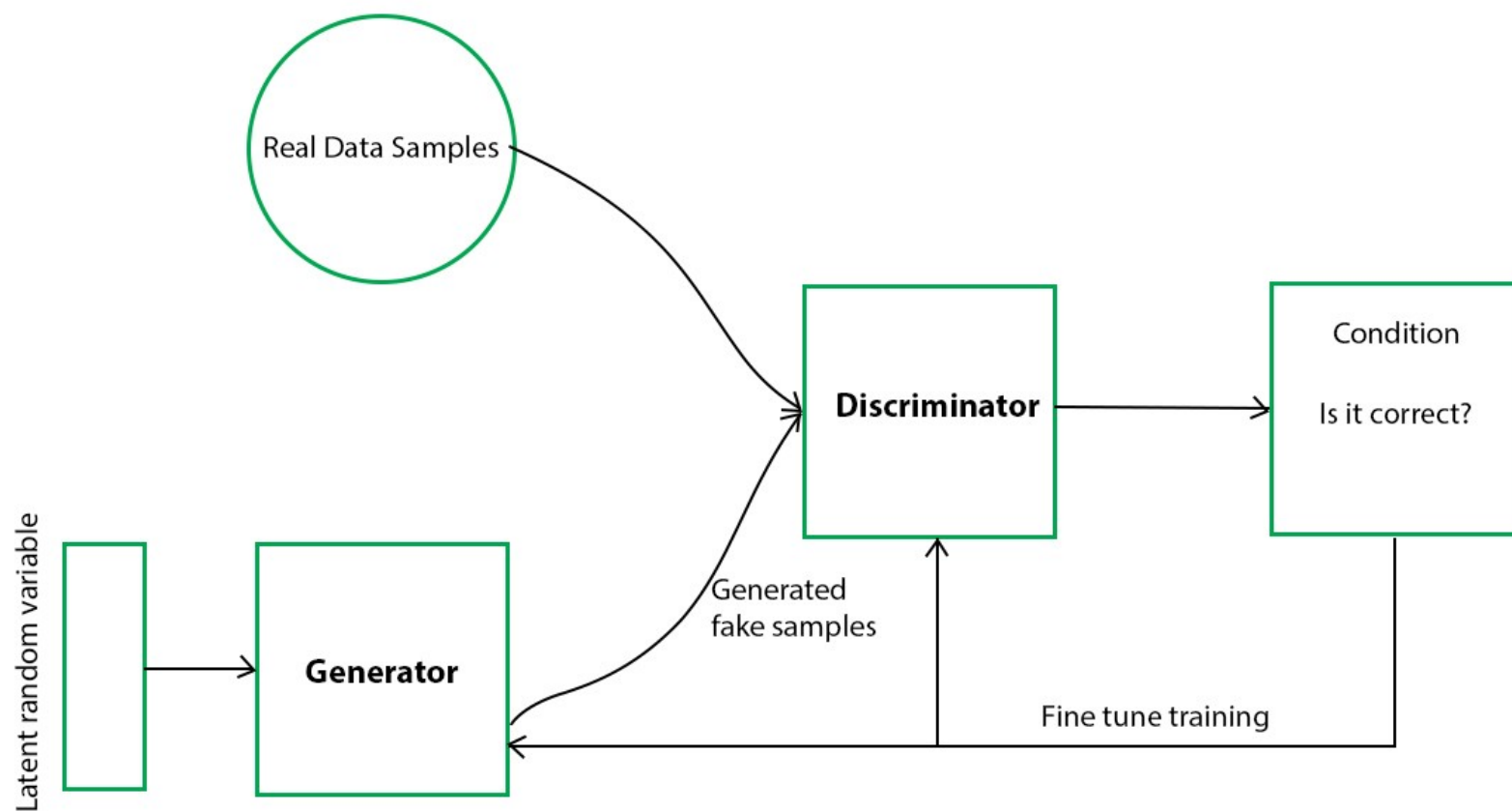
# Вступ

GAN – клас моделей машинного навчання призначених для відтворення зразків з розподілу даних.  
Зазвичай являє собою систему з двох нейронних мереж: дискримінатора та генератора.

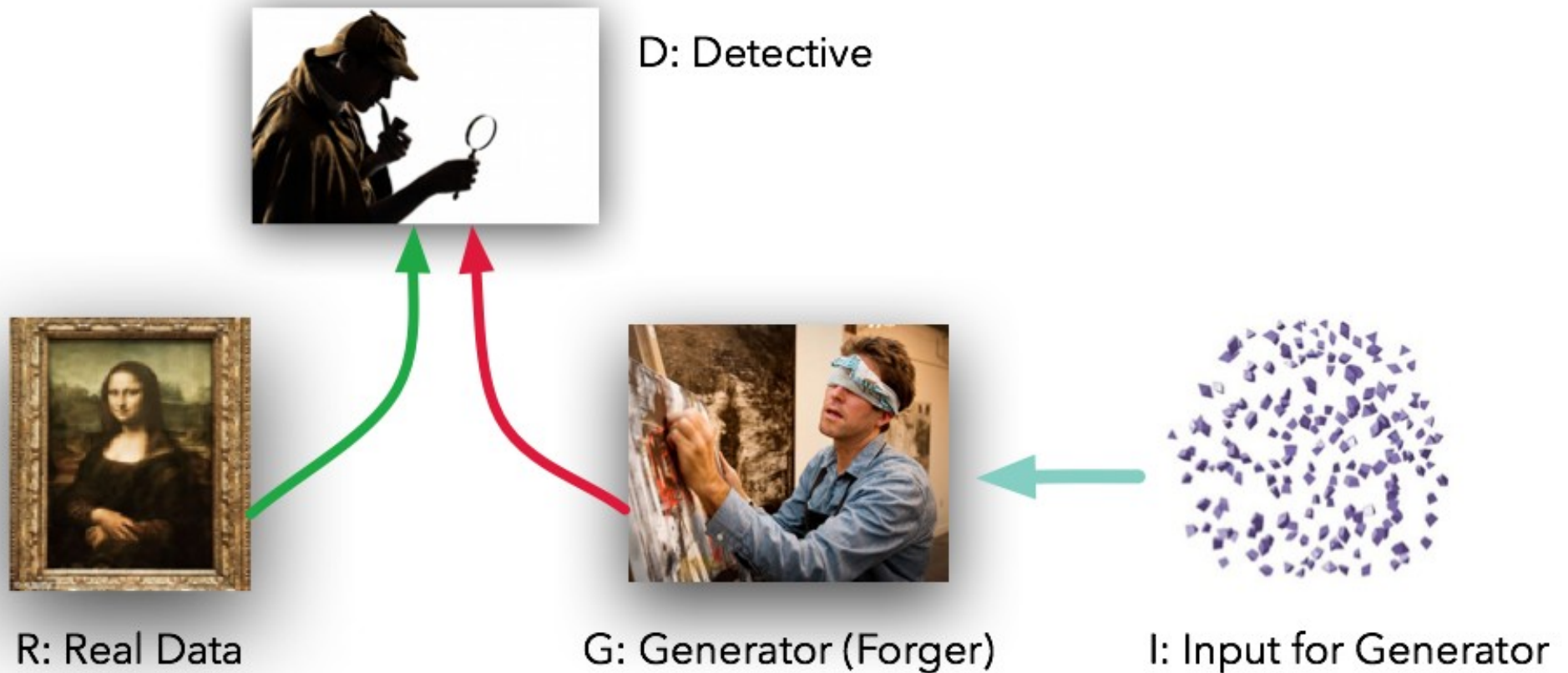
Ключова особливість – використання змагального (adversarial) процесу між цими мережами.



# Загальна схема



# Життєва аналогія



# Математичне формулювання

Позначимо:

- Дані:  $x \in X$ ,  $x \sim p_{data}(x)$ ;
- Вхідний шум:  $z \sim p_{noise}(z)$ ; (latent prior)
- Генератор:  $G(z, \theta_g)$ ; (відображення в простір  $X$ ) характеризується  $p_g(z)$
- Дискримінаатор:  $D(*, \theta_d)$ ; (функція, що повертає ймовірність істинності)

# Математичне формулювання

Тоді процес тренування представляє собою наступну взаємодію генератора та дискримінатора:

$$\min_G \max_D V(D, G) = \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} [\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})} [\log(1 - D(G(\mathbf{z})))].$$

- Тобто, задача D – максимізувати V:  
 $D(\mathbf{x}) \rightarrow 1, D(G(\mathbf{z})) \rightarrow 0$
- У G протилежна задача – мінімізувати V:  
 $D(G(\mathbf{z})) \rightarrow 1 \quad (D(\mathbf{x}) \rightarrow 0)$



# Математичне формулювання

В оригінальній статті доводиться, що  $p_g$   
*сходиться до  $p_{data}$ , та відповідно  $D(G(z)) \rightarrow 1/2$ .*

# Алгоритм навчання

**for** number of training iterations **do**

**for**  $k$  steps **do**

- Sample minibatch of  $m$  noise samples  $\{z^{(1)}, \dots, z^{(m)}\}$  from noise prior  $p_g(z)$ .
- Sample minibatch of  $m$  examples  $\{x^{(1)}, \dots, x^{(m)}\}$  from data generating distribution  $p_{\text{data}}(x)$ .
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^m \left[ \log D(x^{(i)}) + \log (1 - D(G(z^{(i)}))) \right].$$

**end for**

- Sample minibatch of  $m$  noise samples  $\{z^{(1)}, \dots, z^{(m)}\}$  from noise prior  $p_g(z)$ .
- Update the generator by descending its stochastic gradient:

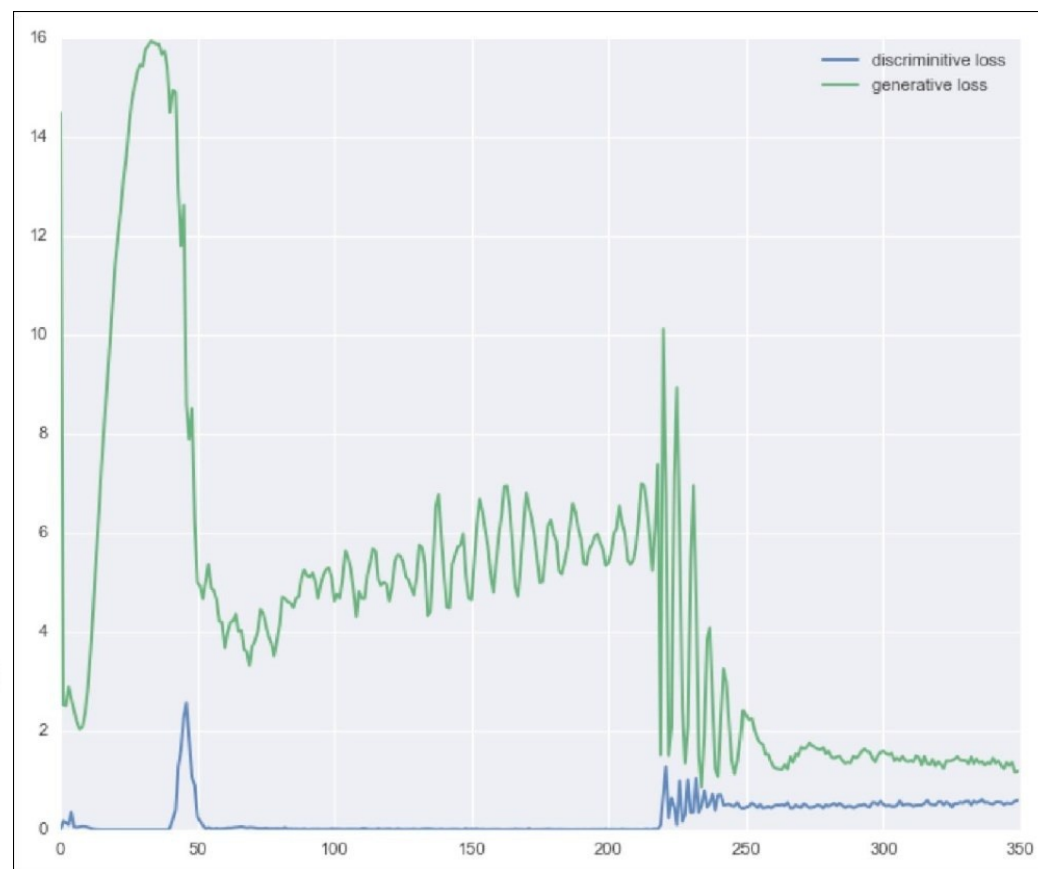
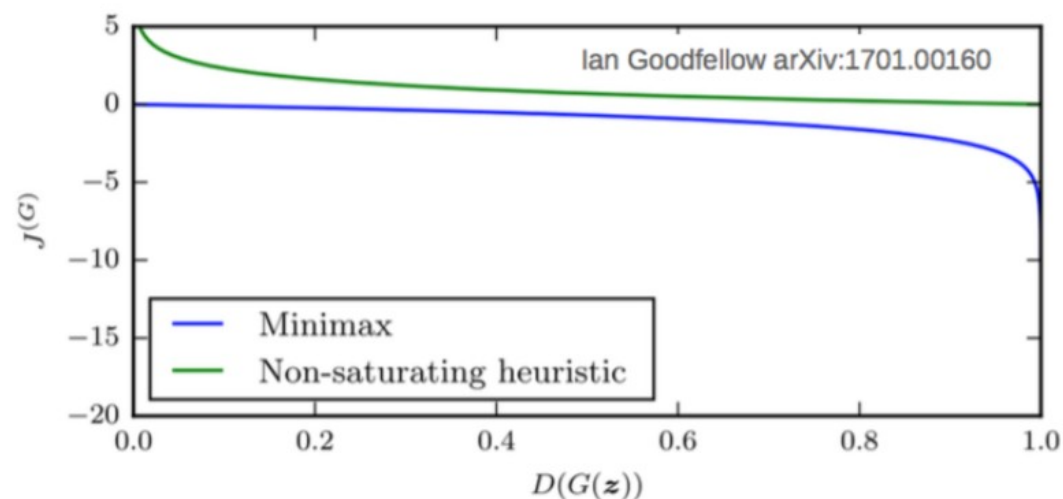
$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^m \log (1 - D(G(z^{(i)}))).$$

**end for**



# Проблема насичення

*На початку навчання дискримінатор вчиться значно швидше, що створює проблему зникаючого градієнта в генераторі.*



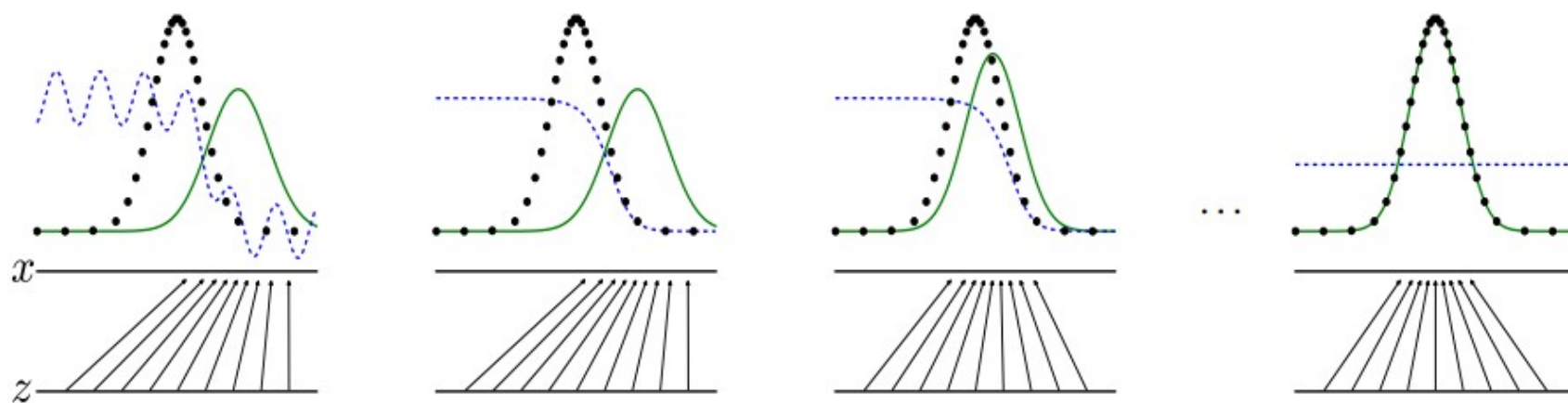
# Проблема насыщения

Тому на початкових етапах навчання (доки генератор не стане більш-менш адекватним) роблять:

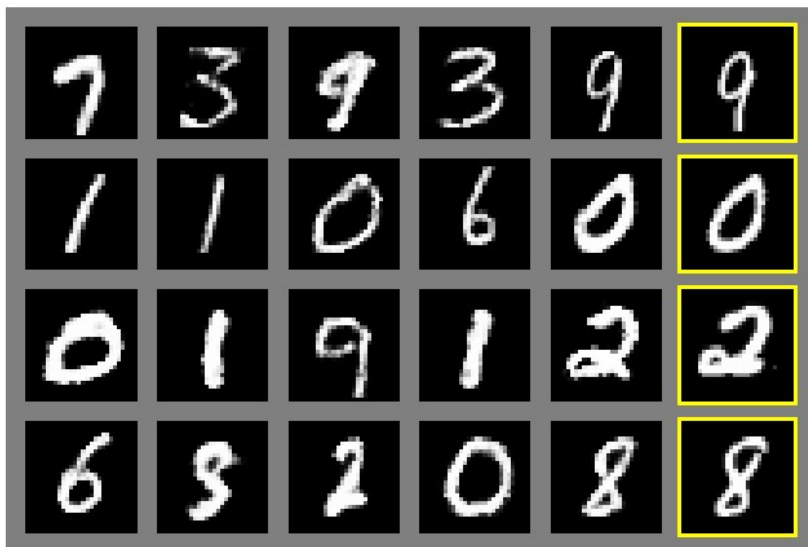
$$\text{minimize } \theta_G \{ \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z; \theta_G); \theta_D))] \}$$

$$\text{maximize } \theta_D \{ \mathbb{E}_{z \sim p_z(z)} [\log D(G(z; \theta_G); \theta_D)] \}$$

# Алгоритм навчання



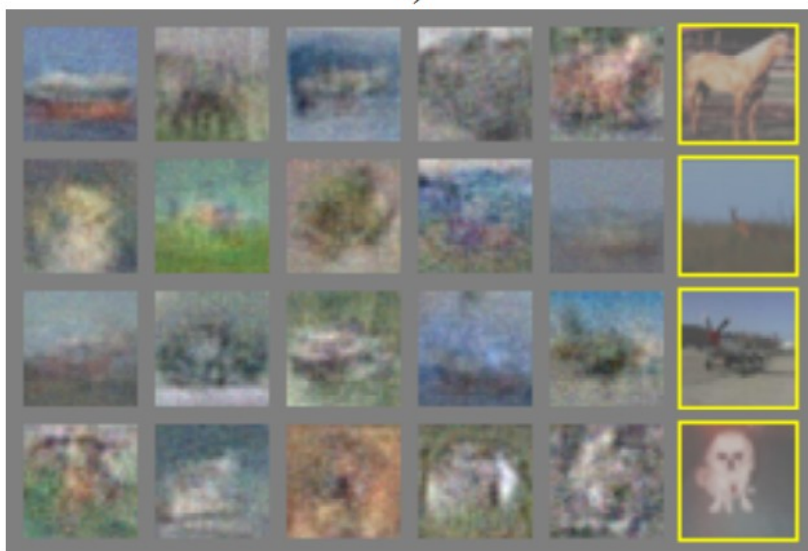
# Застосування: генерування зображень



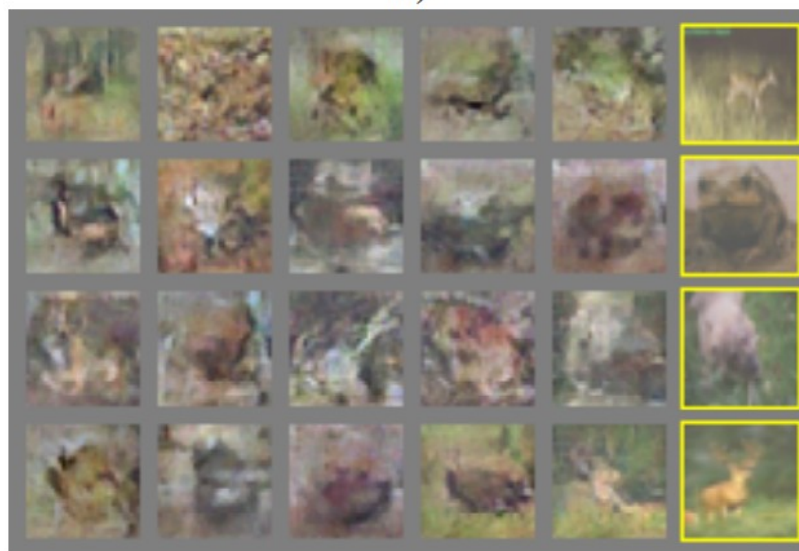
a)



b)



c)



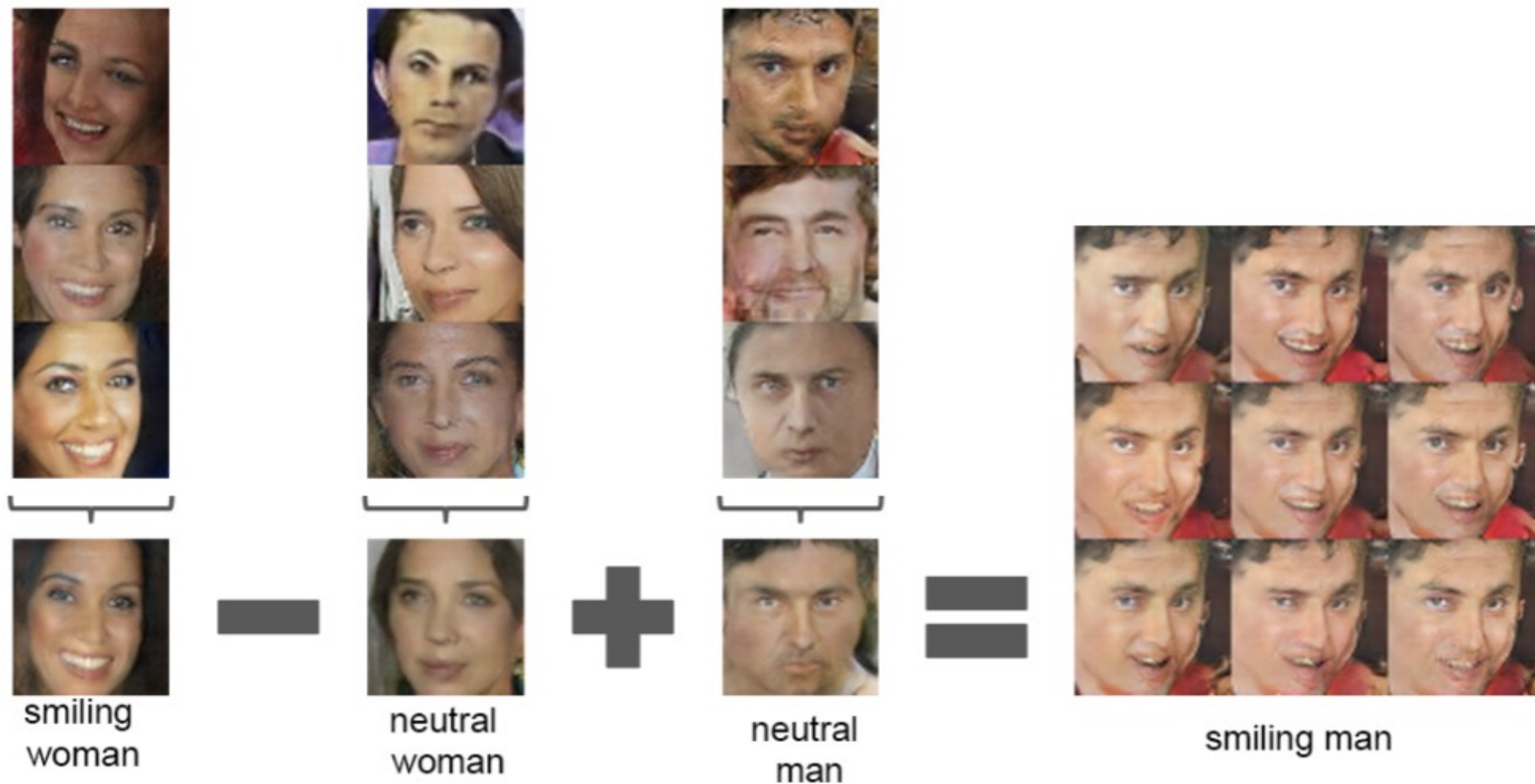
d)



# Застосування: генерування зображень



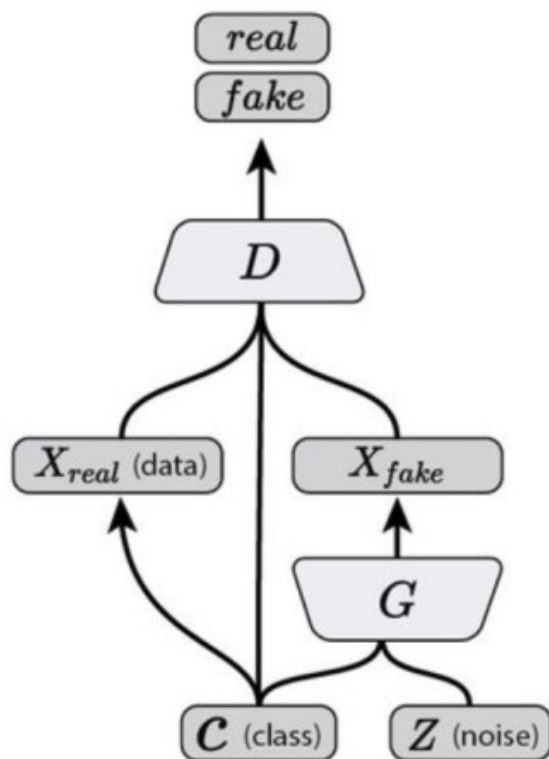
# Векторні операції в латентному просторі



# Розширення GAN

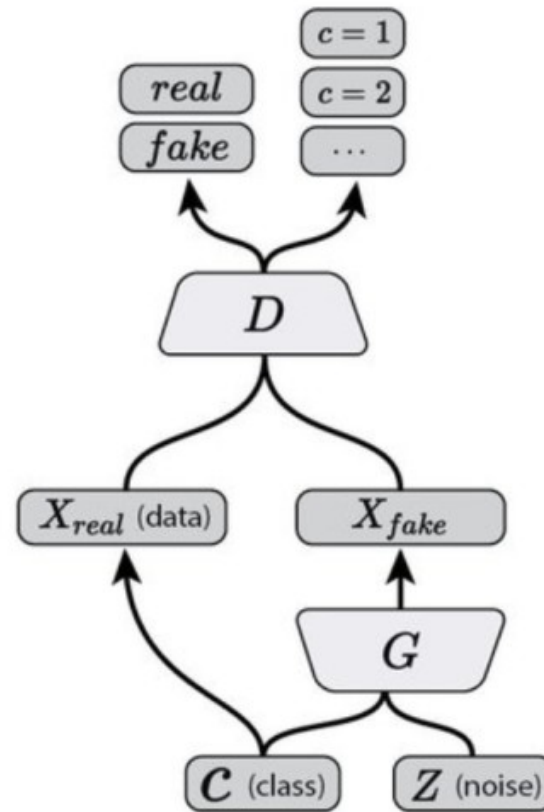
## Conditional GAN

(Mirza & Osindero, 2014)



## Auxiliary Classifier GAN

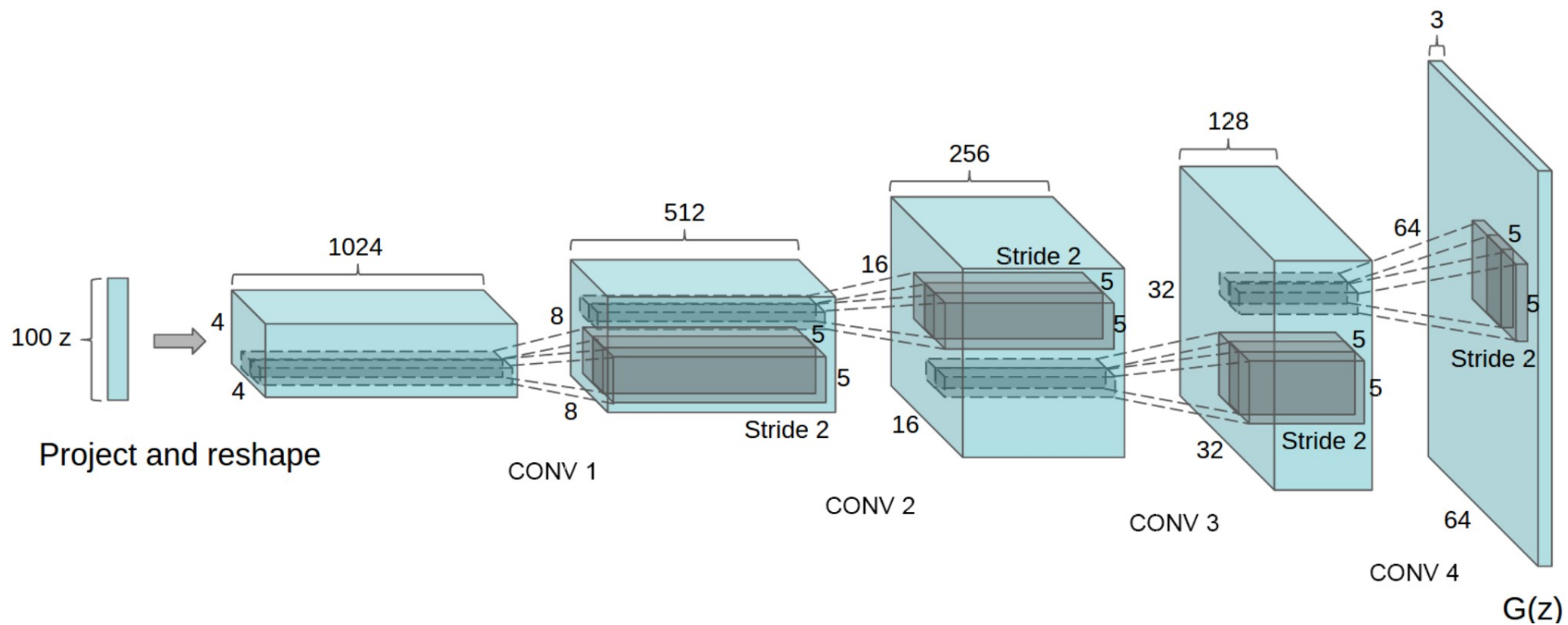
(Odena, et al., 2016)





# Розширення GAN

## Deep Convolutional GAN

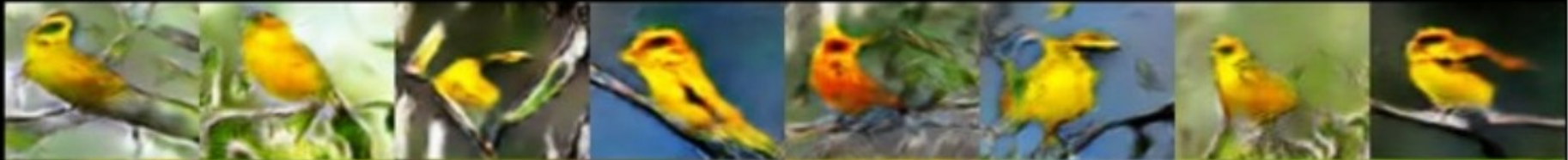




# Розширення GAN

this bird has a bright yellow body, with brown on its crown and wings.

Stage-I



Stage-II



A white bird with a black crown and red beak

Stage-I



Stage-II



this bird has red breast and yellow belly