

SIMULACIÓN DE SUCECOS DE COLISIÓN UTILIZANDO REDES GENERATIVAS ADVERSARIAS

Miguel Ángel Hoyo Abascal

Director: Pablo Martínez Ruiz del Árbol

Co-director: Sergio Sánchez Cruz

Tabla de contenidos

Introducción

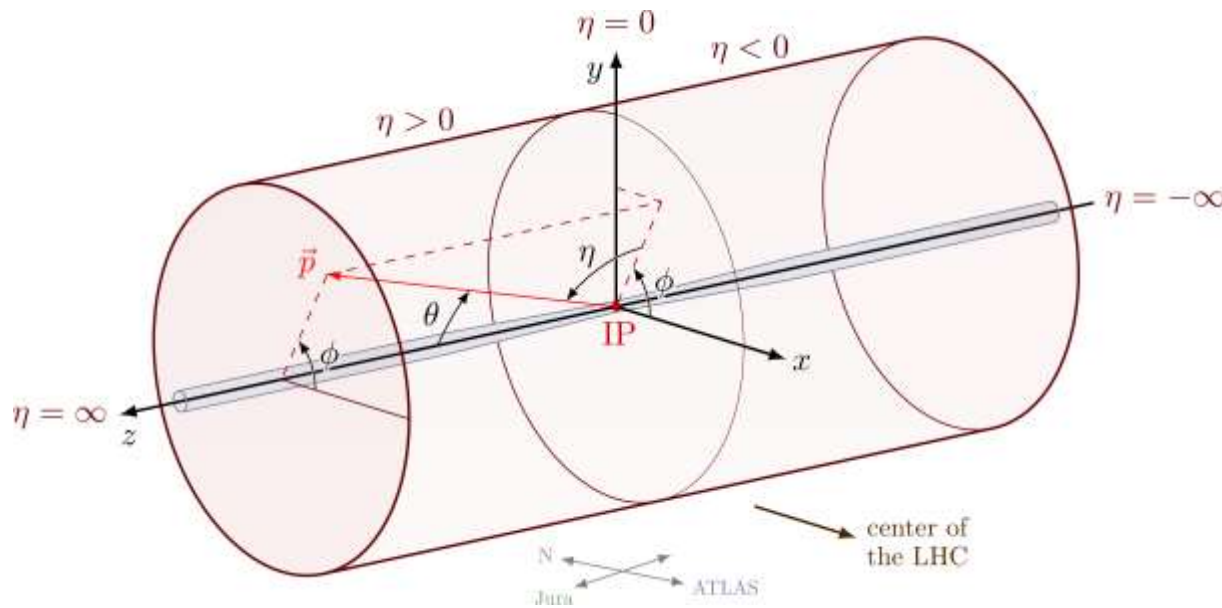
Redes Neuronales (WGAN)

Entrenamiento y resultados

Conclusiones

Introducción

Simulaciones en física de altas energías



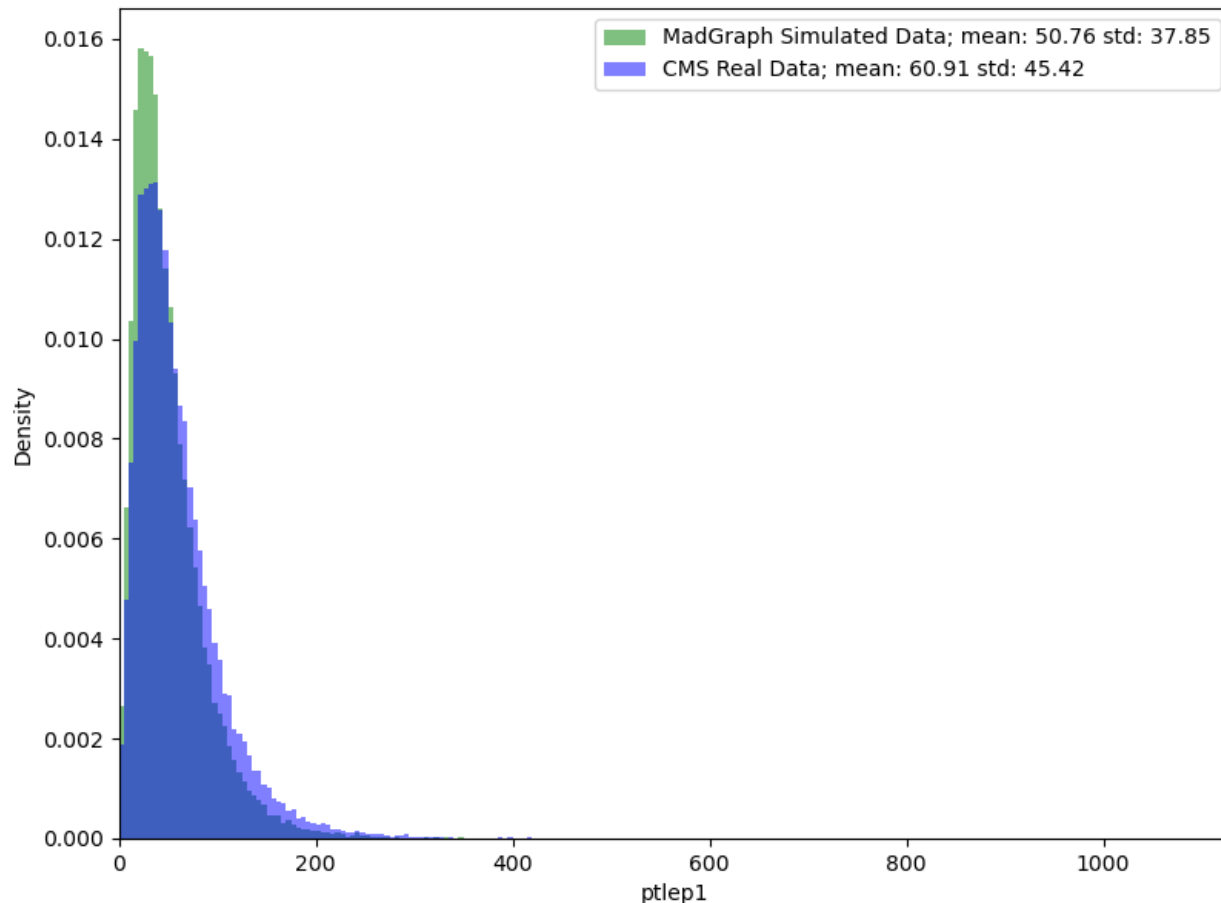
- Magnitudes objetivo:

- $p_T \in [0, +\infty]$
- $\phi \in [-\pi, +\pi]$
- $\eta \in [-\infty, +\infty]$; $\eta = -\ln \tan \frac{\theta}{2}$

- Fuentes de inexactitudes:

- Aproximación de la sección eficaz
- Modelo tridimensional del CMS
- Fallo de los subdetectores
- Procesos de nueva física

Producción de quarks top-antitop

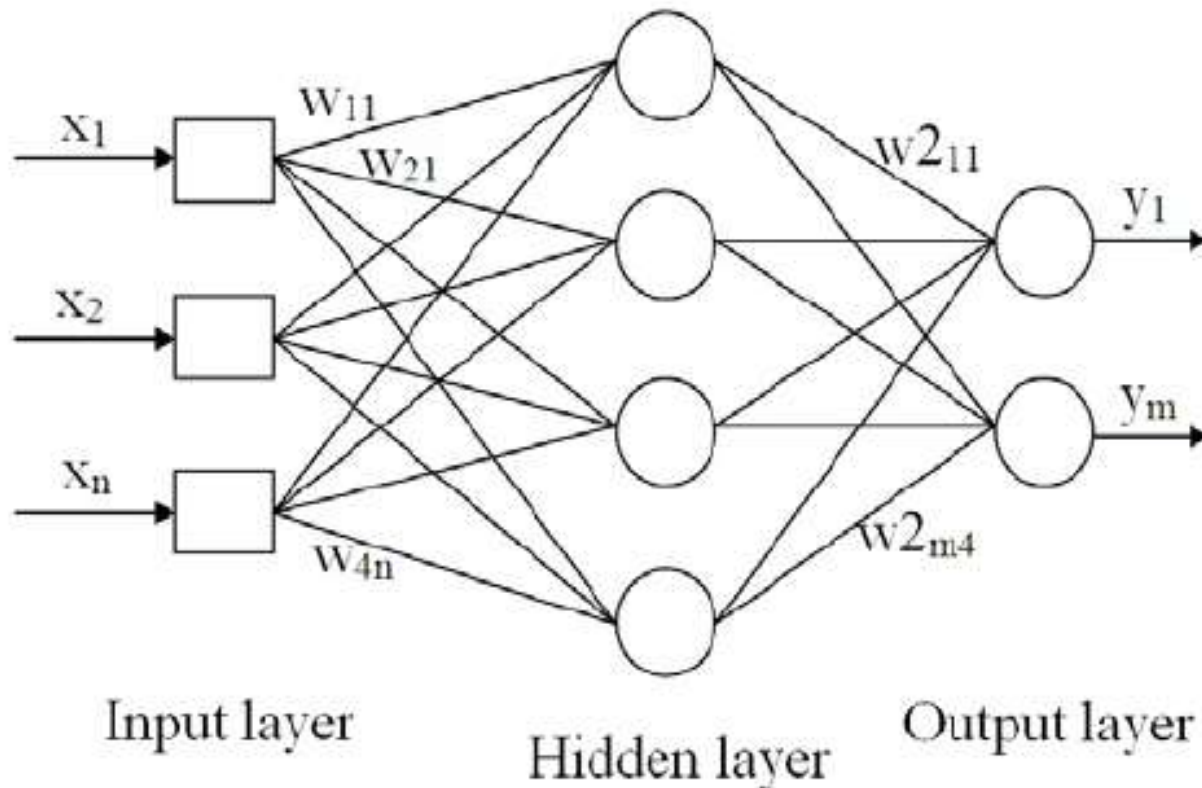


Objetivos

- Corrección de las distribuciones
- Partículas:
 - 2 leptones
 - 2 *b-jets*
- MET:
 - *Missing Transverse Energy*
 - Debido a leyes de conservación

Redes Neuronales (WGAN)

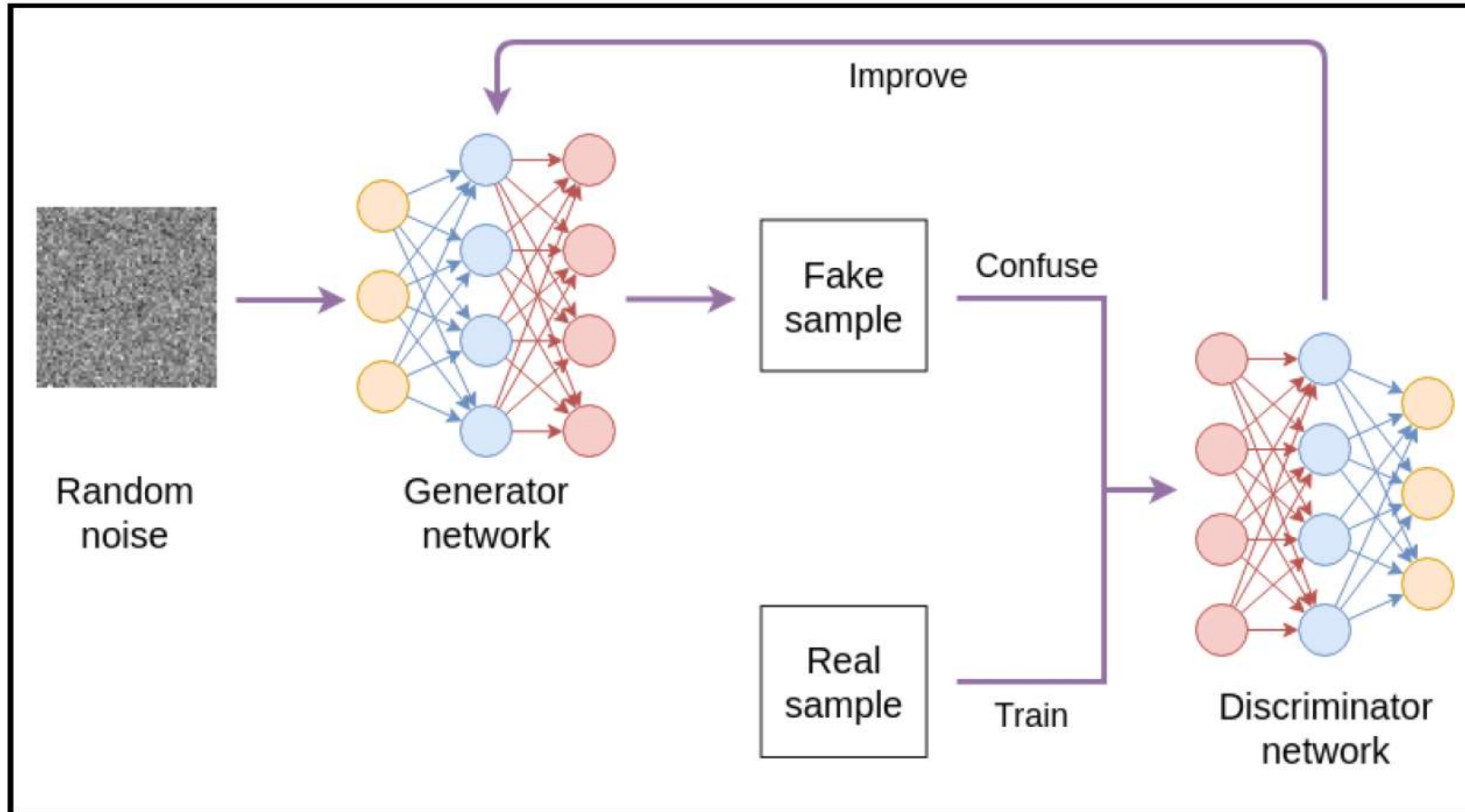
Esquema de una red neuronal



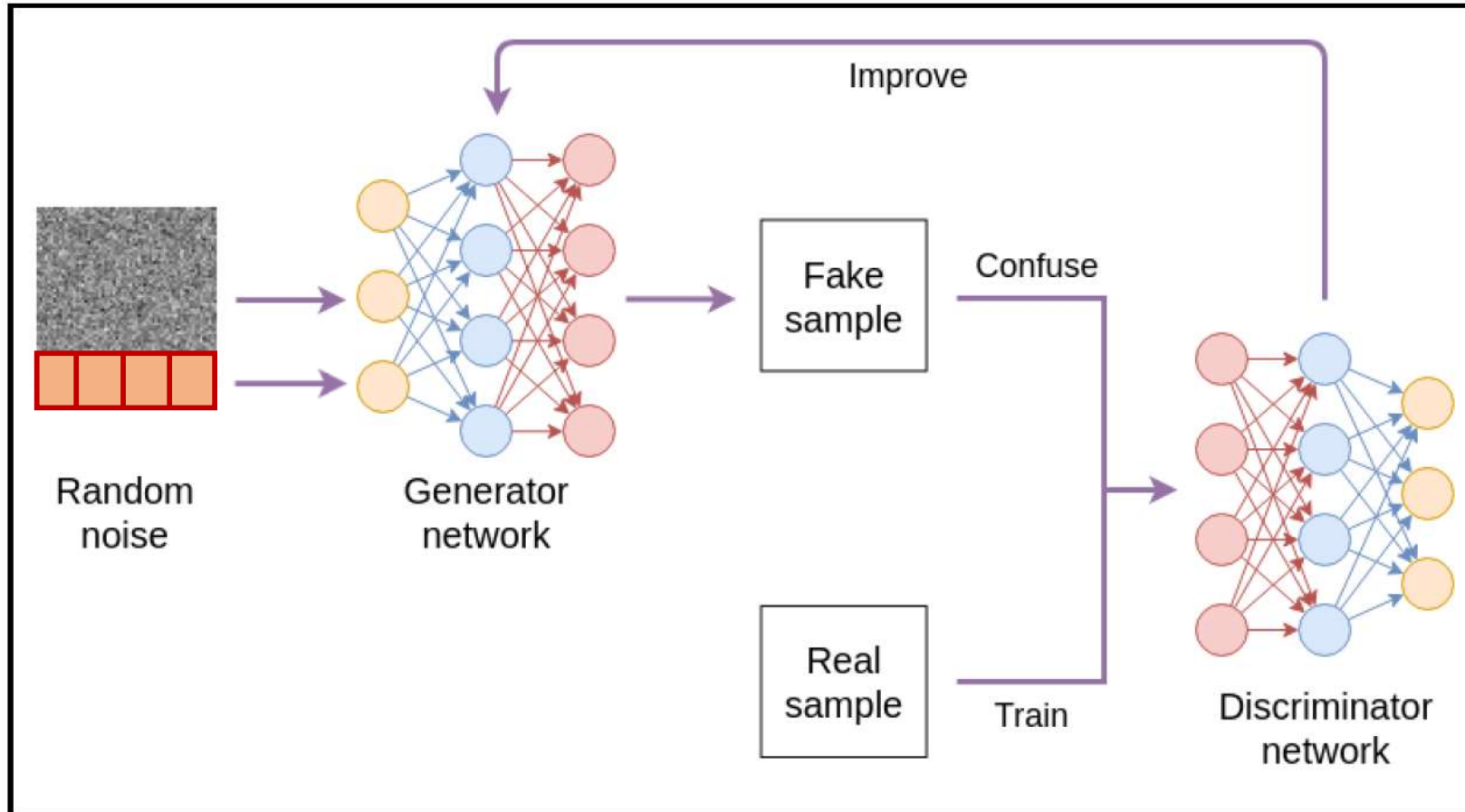
Componentes

- *Input layer*
 - $\{x_n\} \rightarrow \text{inputs}$
- *Hidden layers*
 - $\{w_{ij}\} \rightarrow \text{pesos}$
- *Output layer*
 - $\{y_m\} \rightarrow \text{outputs}$

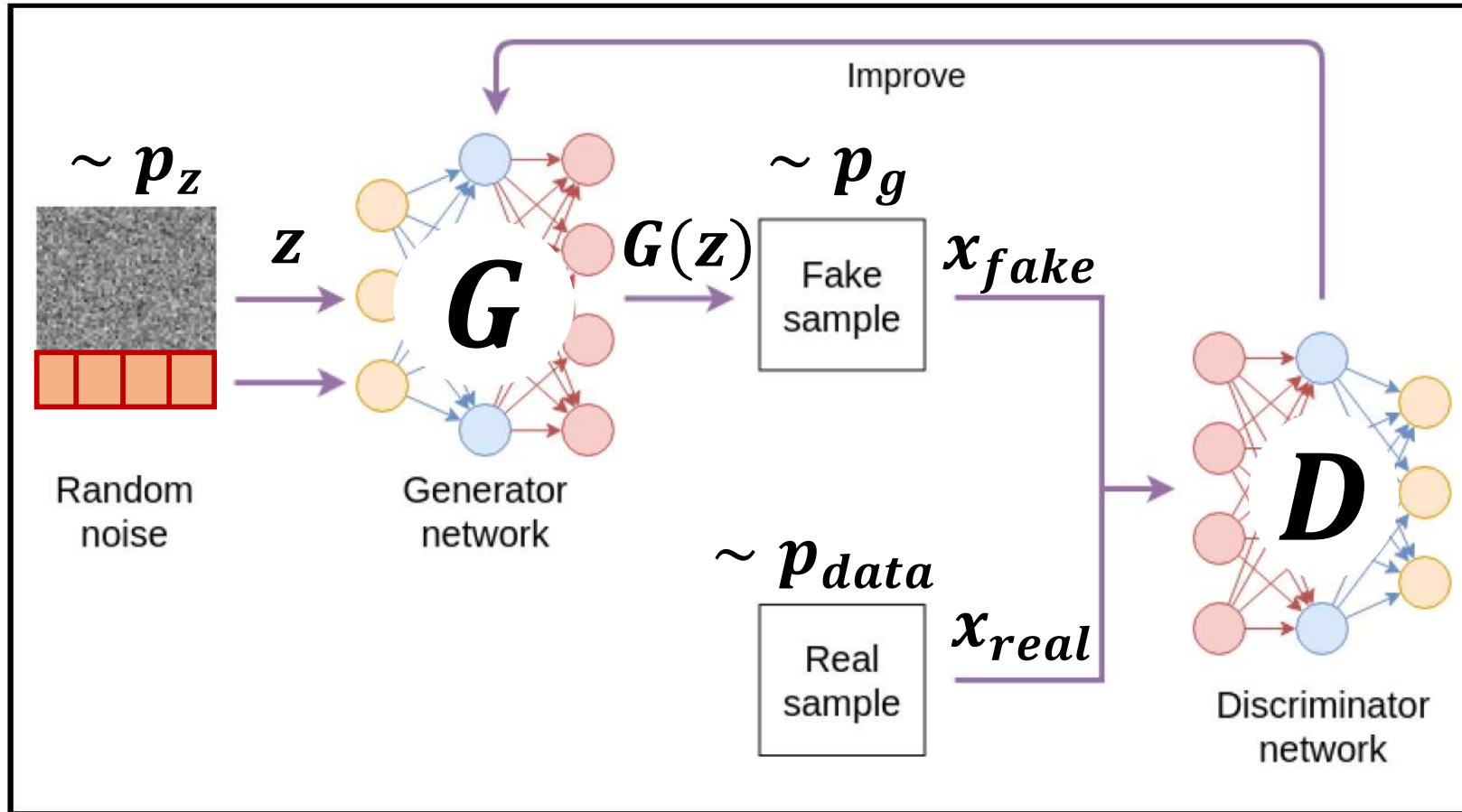
Redes generativas adversarias (GAN)



Vector de información



Redes generativas adversarias (GAN)

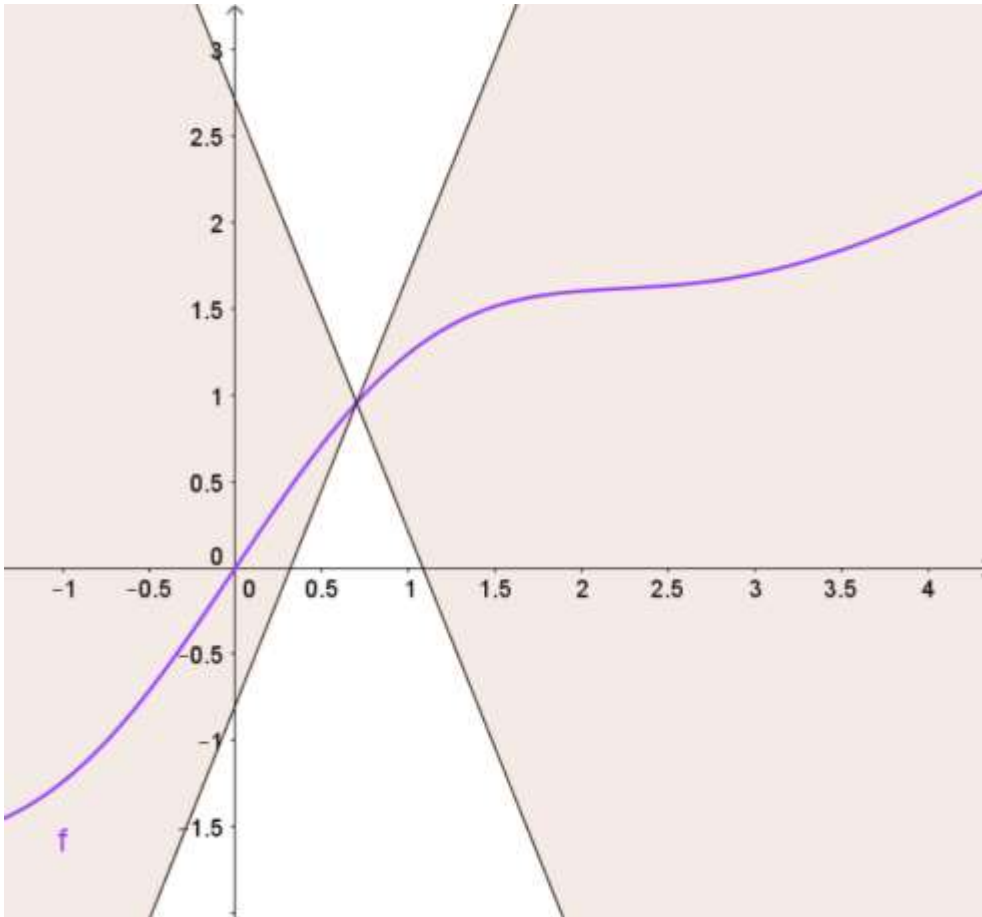


Función objetivo Wasserstein GAN

- Basado en:
 - Distancia de Wasserstein
 - Continuidad de Lipschitz
- Propósito de D :
 - Calcular la distancia de Wasserstein

$$\min_G \max_{w \in W} V(G, D_w) = \mathbb{E}_{x \sim p_{data}} [D_w(x)] - \mathbb{E}_{z \sim p_z} [D_w(G(z))]$$

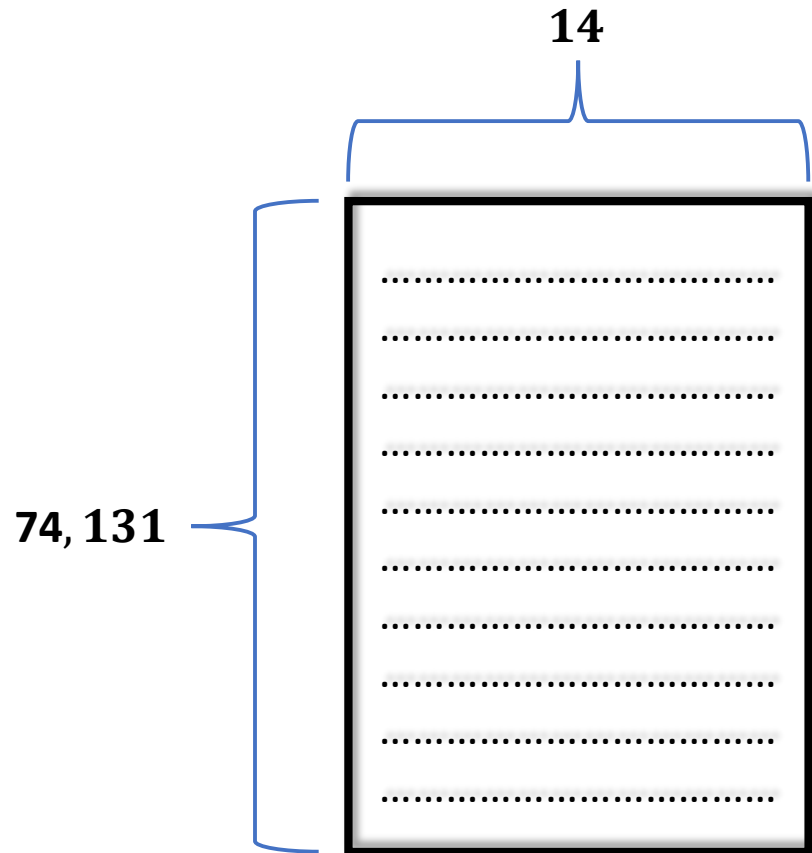
Restricción de Lipschitz



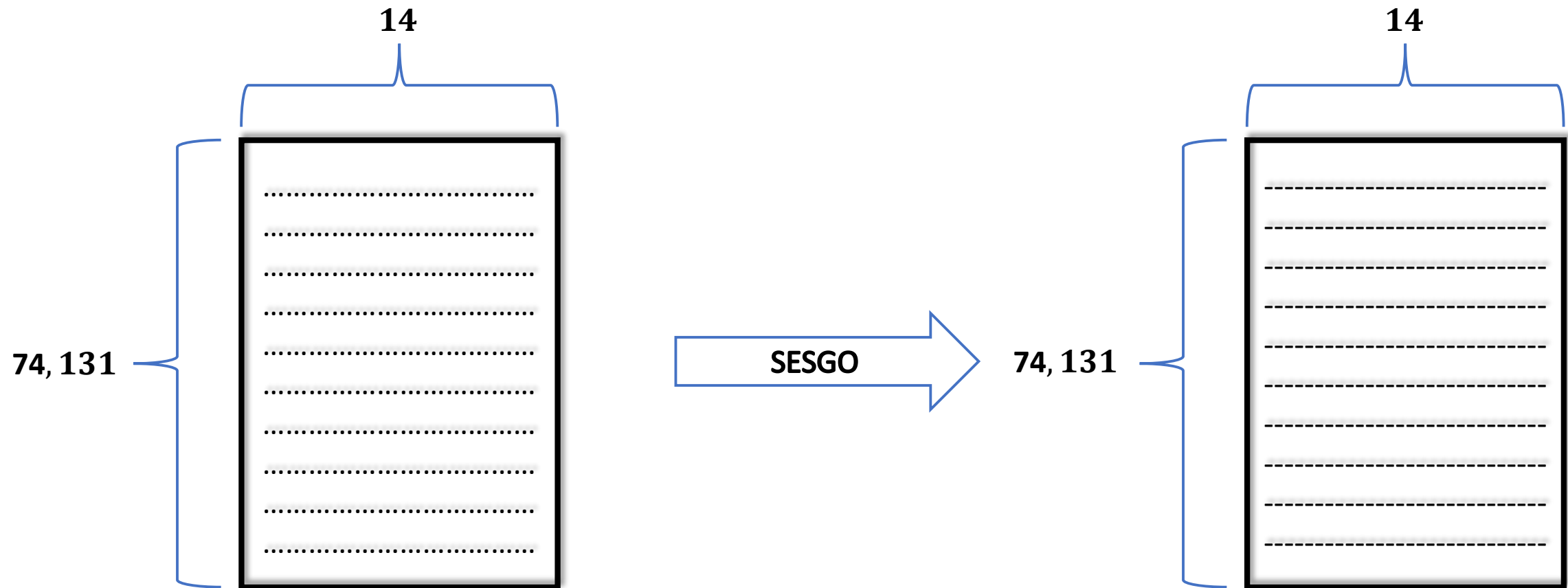
- Condición sobre la derivada
 - $|f(x_1) - f(x_2)| \leq K |x_1 - x_2|$
- Implementaciones:
 - Confinando los parámetros en $D \rightarrow c$
 - Añadiendo un término de *penalty* $\rightarrow \lambda$

Entrenamiento y resultados

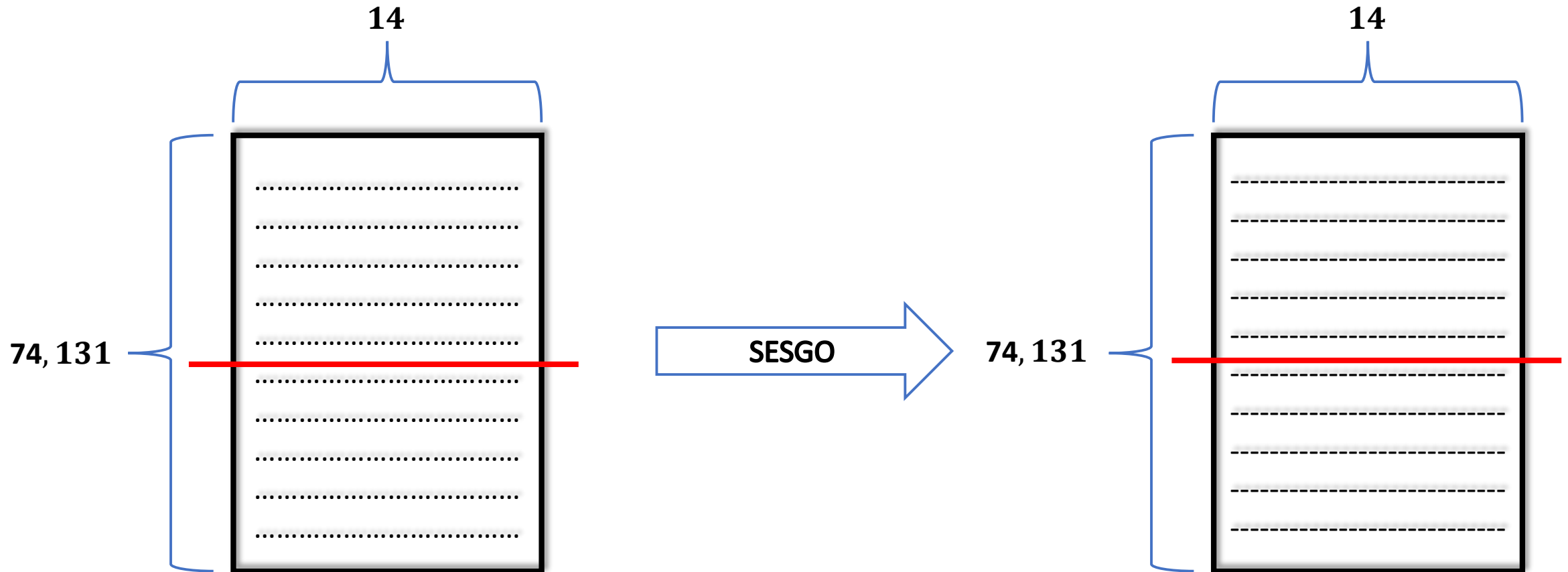
División del *dataset*



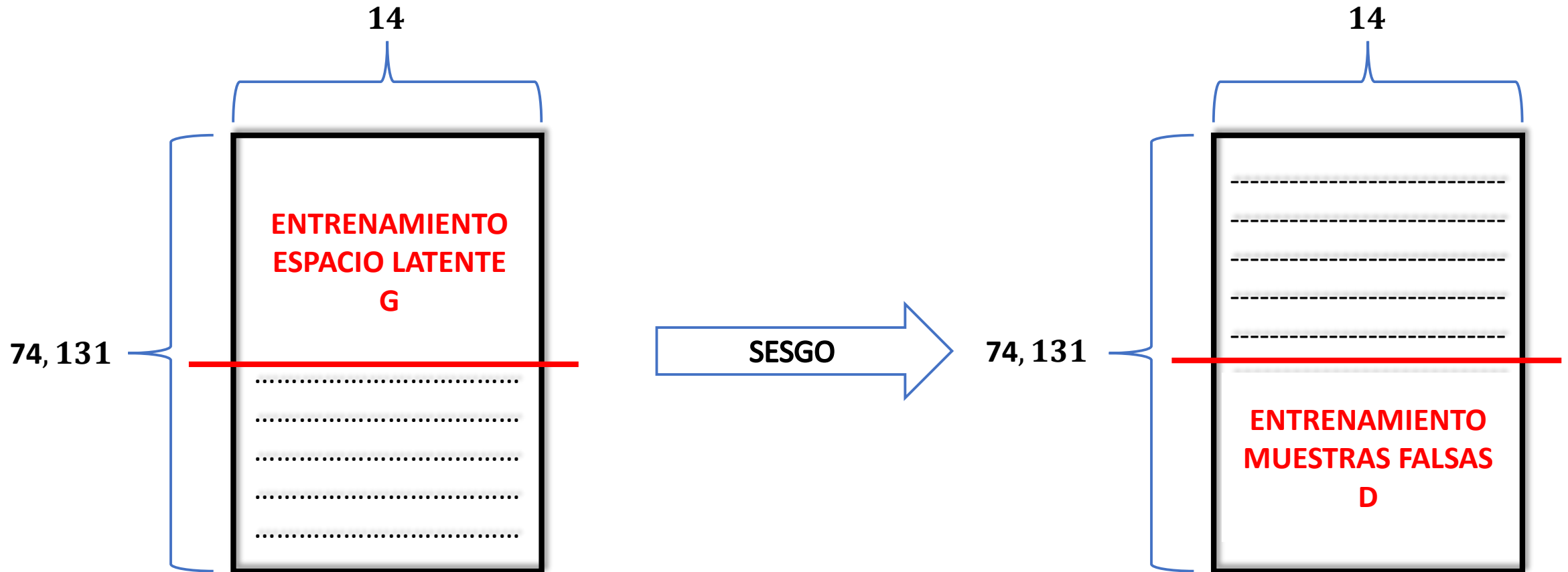
División del *dataset*



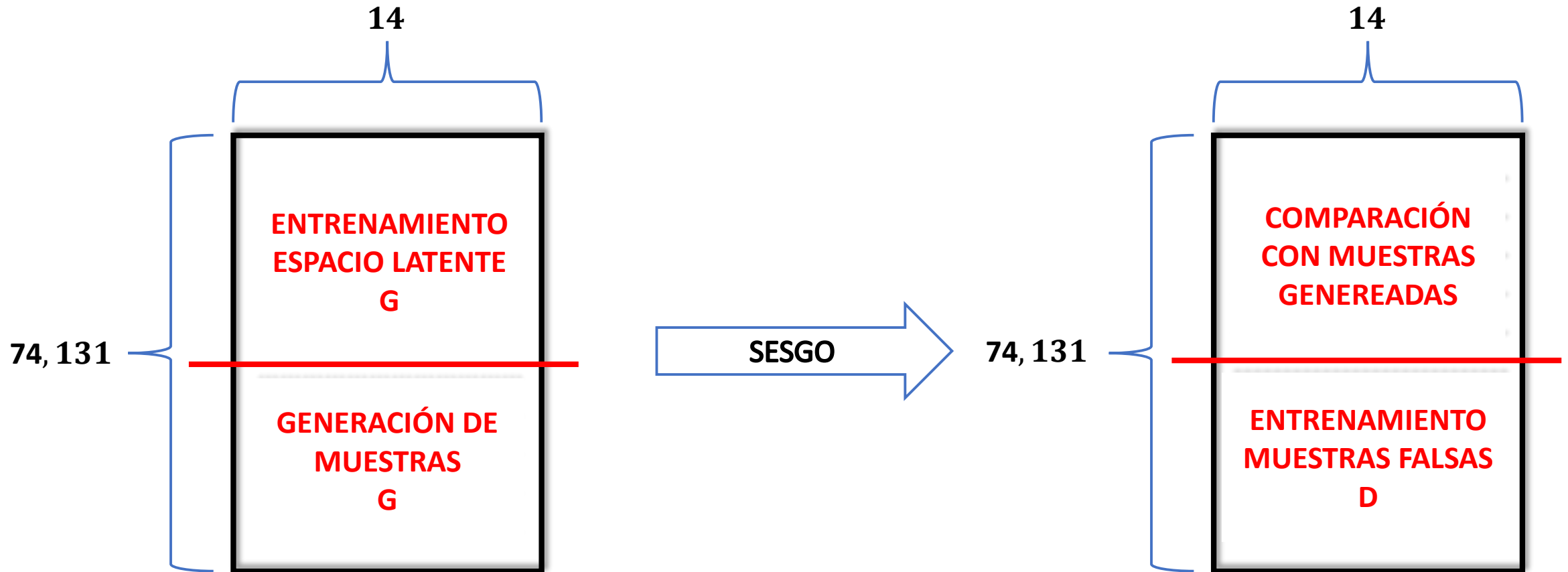
División del *dataset*



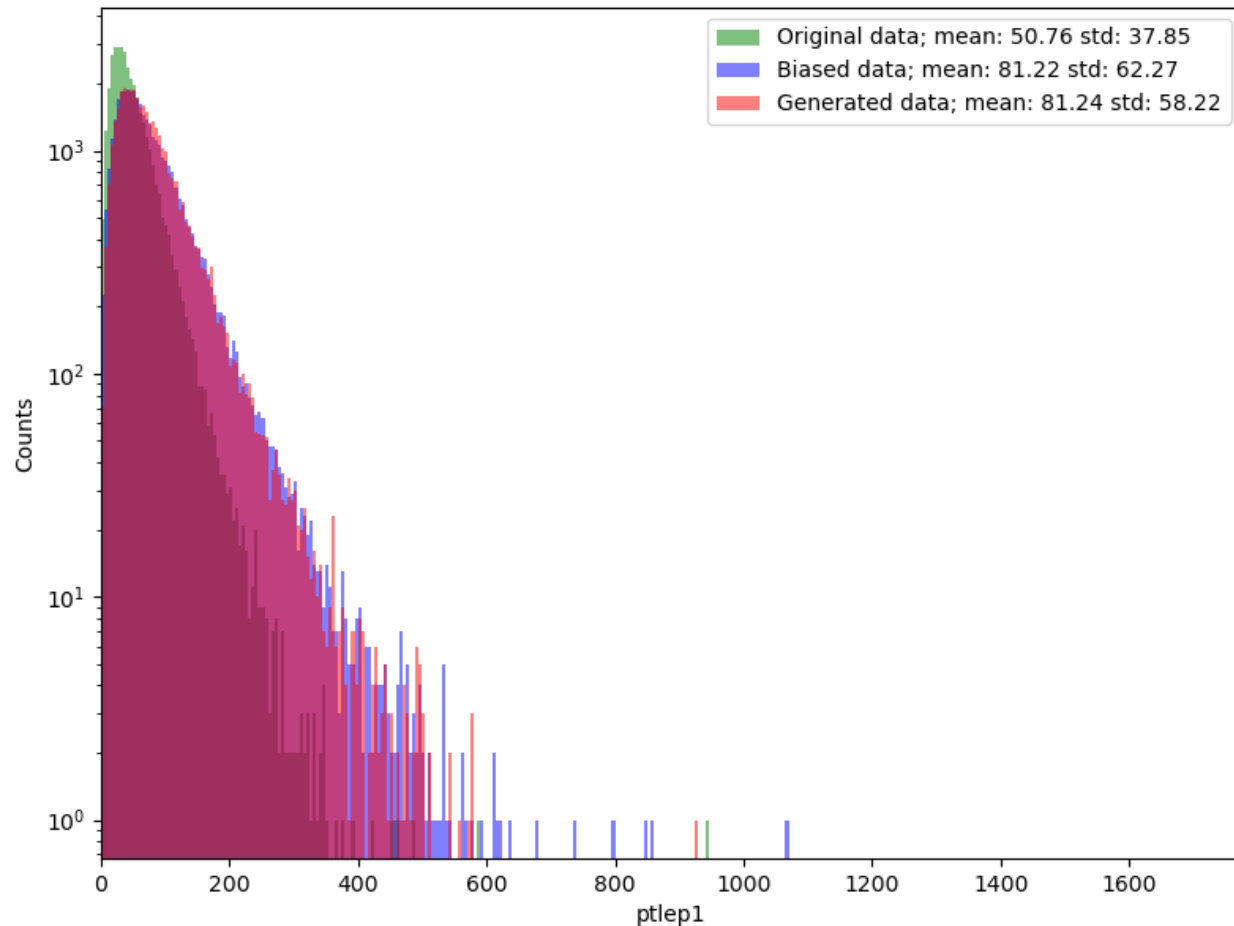
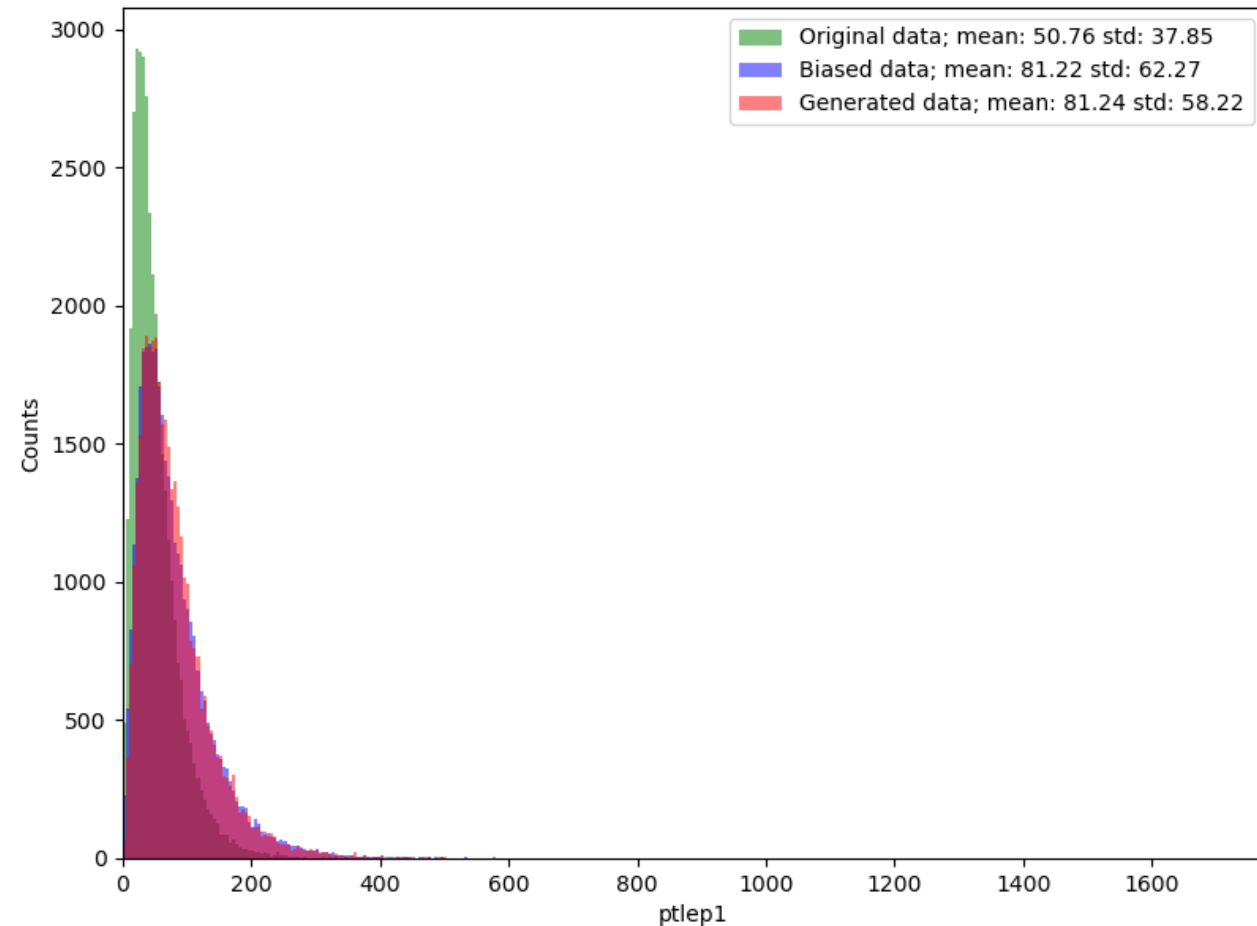
División del *dataset*



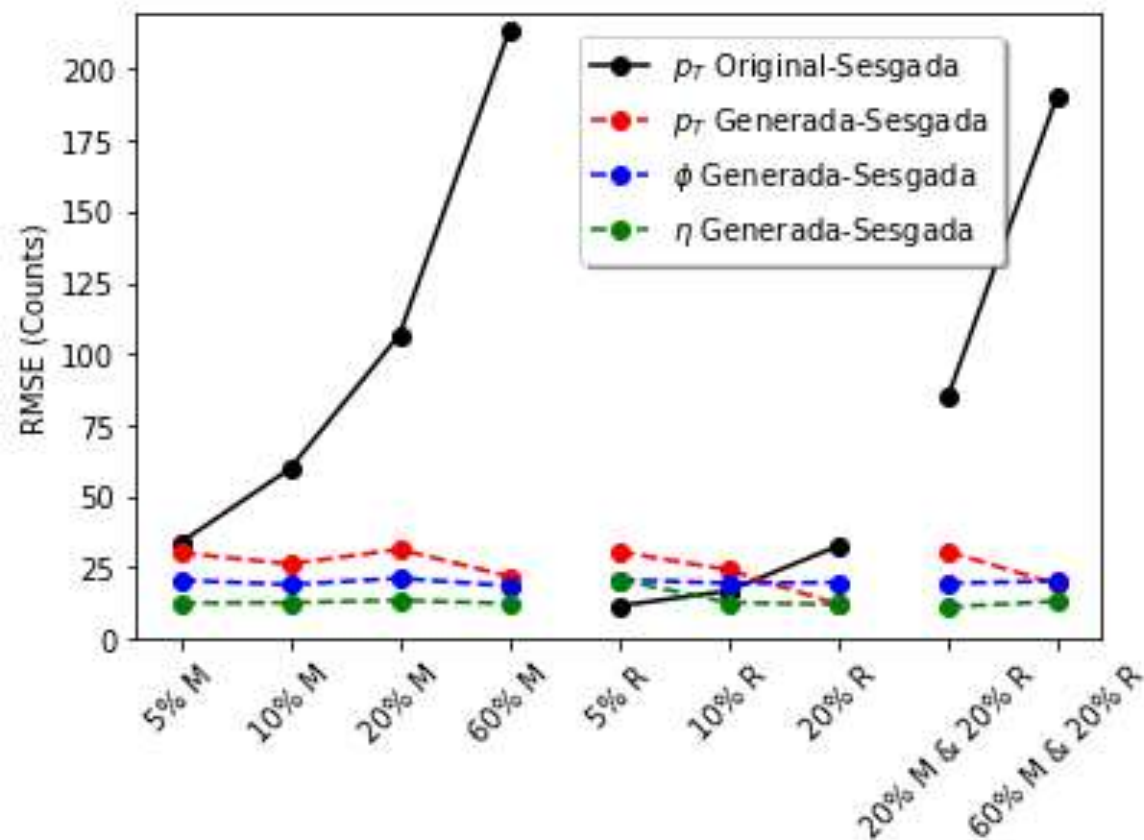
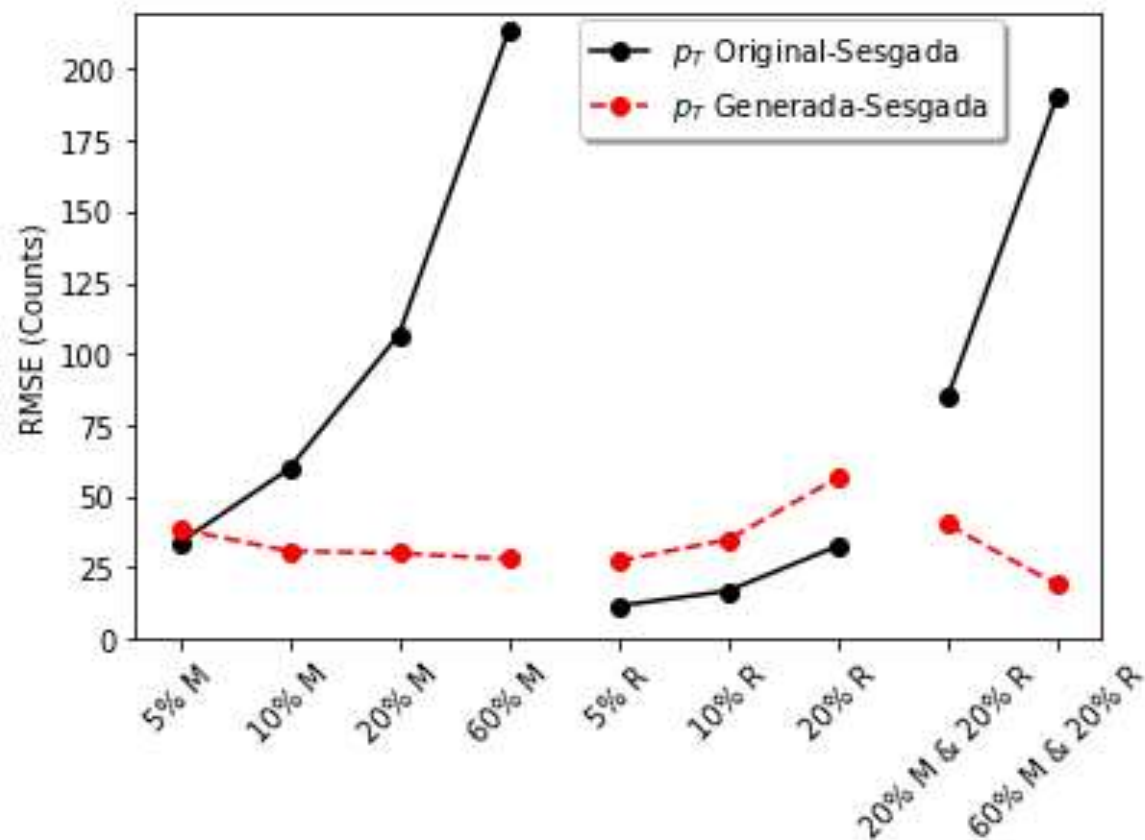
División del *dataset*



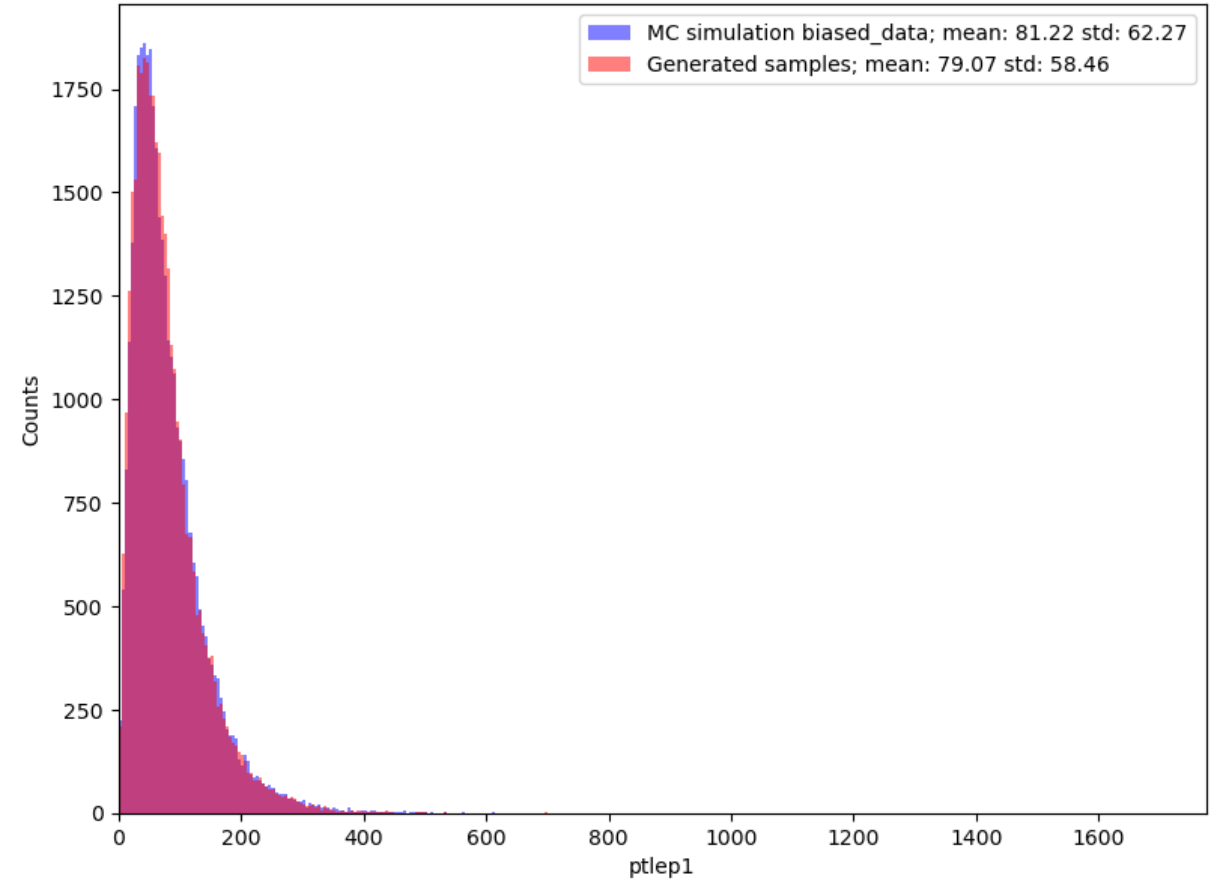
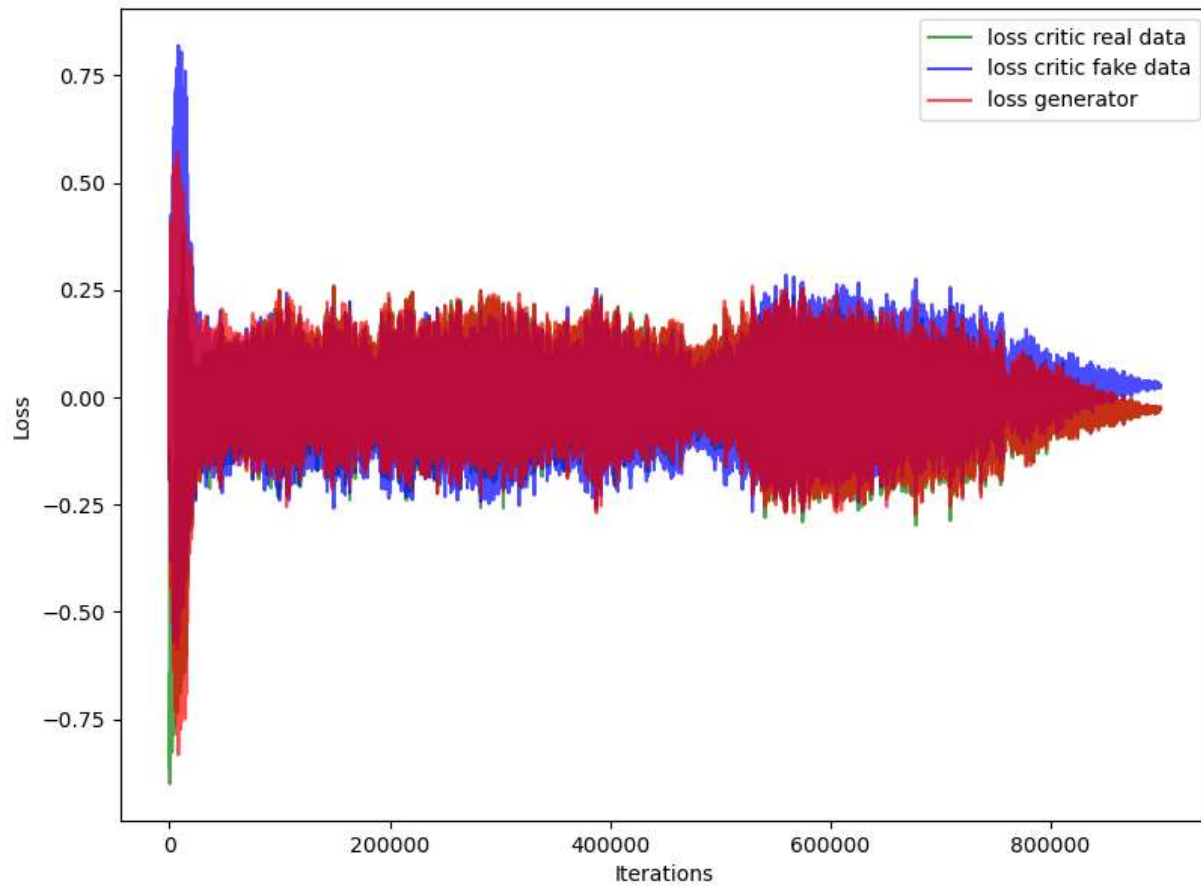
Comparación entre muestras



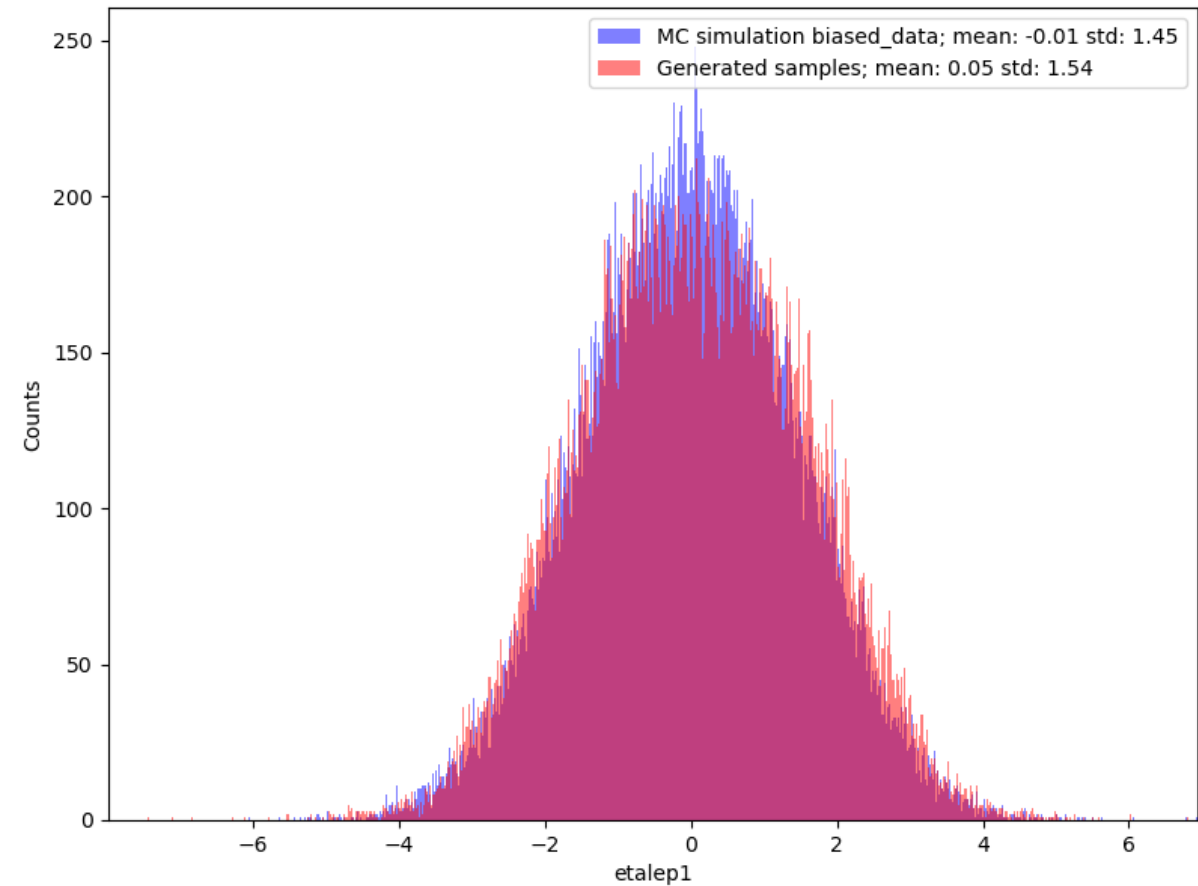
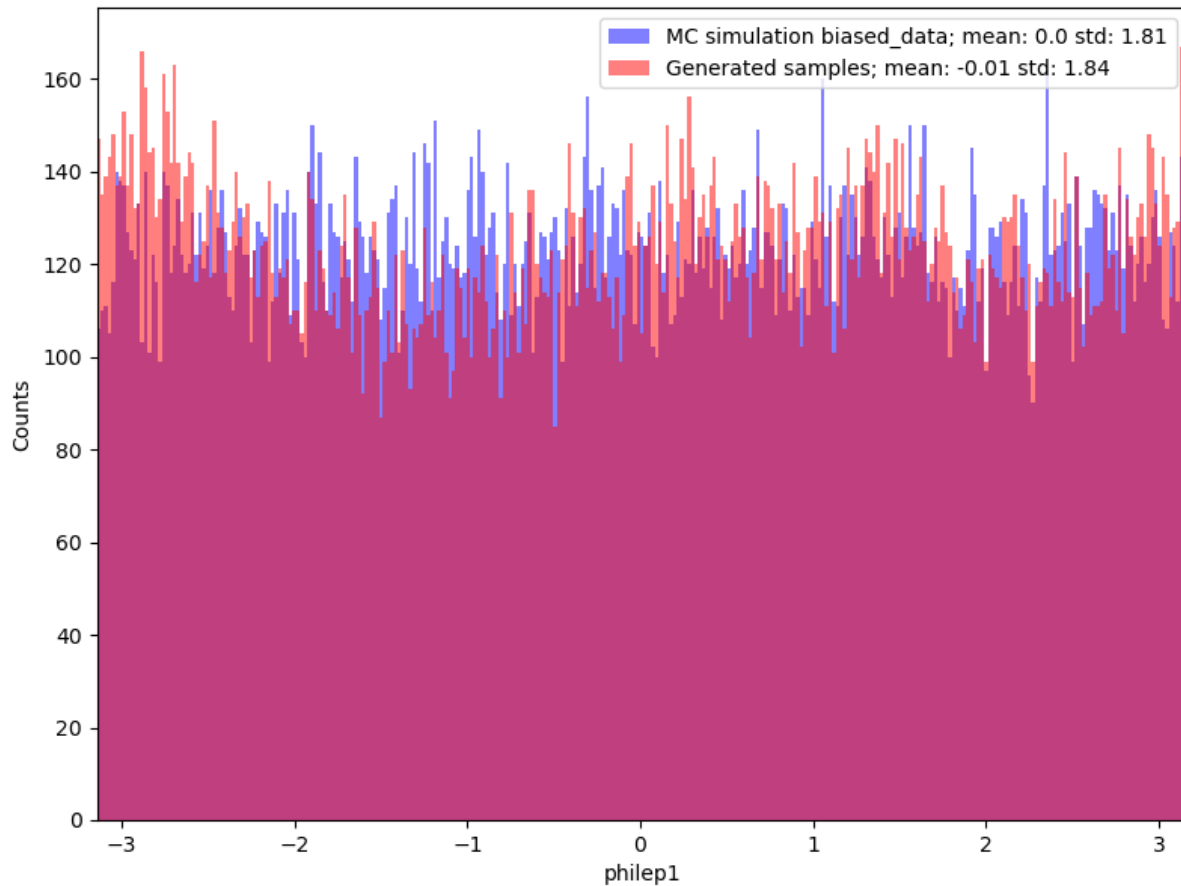
WGAN con 1 y 3 variables



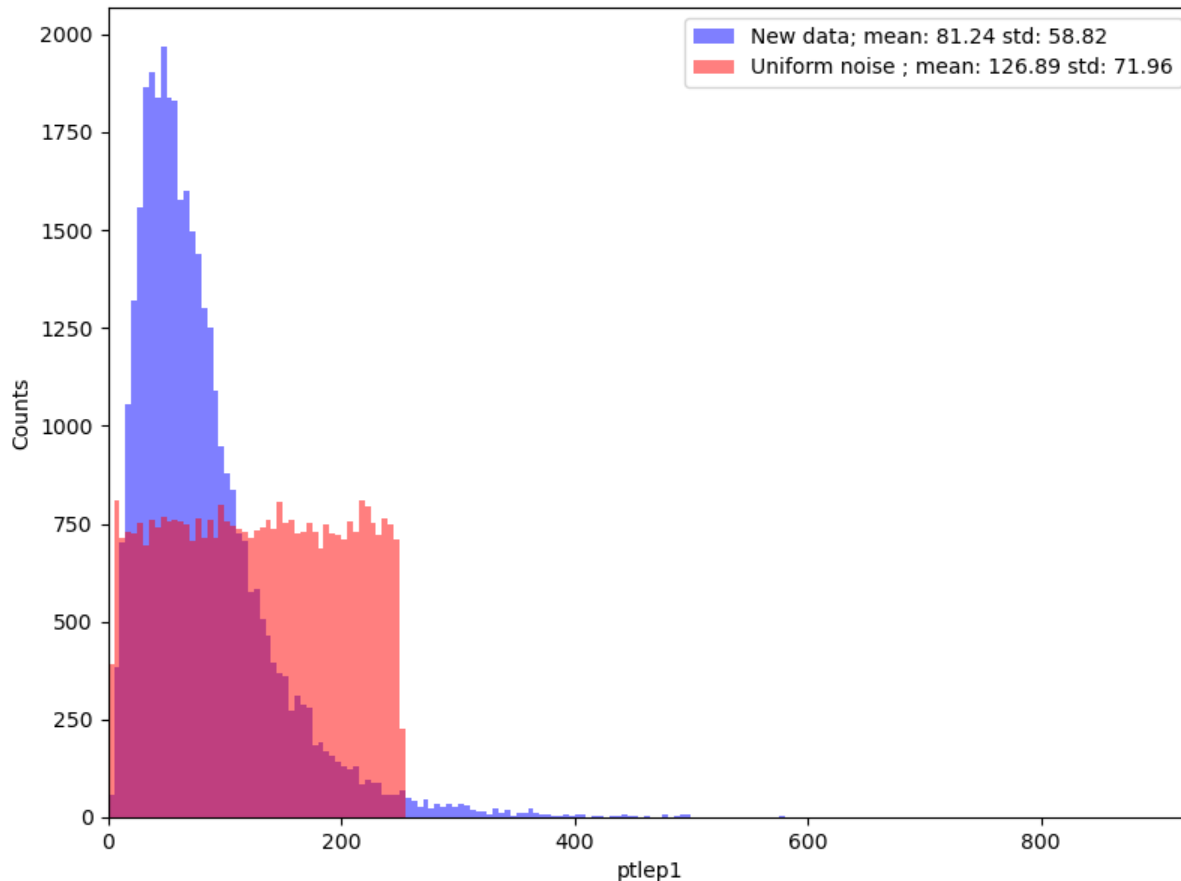
Modelo WGAN con 14 variables



Modelo WGAN con 14 variables



Capacidad de la WGAN



- Sesgo en la red del 60% en la media
- Espacio latente en el entrenamiento:
 - 10 canales de ruido uniforme
 - 1 canal de información (p_T)
- Espacio latente nuevo:
 - 10 canales de ruido uniforme
 - 1 canal de ruido uniforme $\in [0,160]$
- Muestra resultante:
 - Distribución uniforme
 - Rango $\in [0,256]$

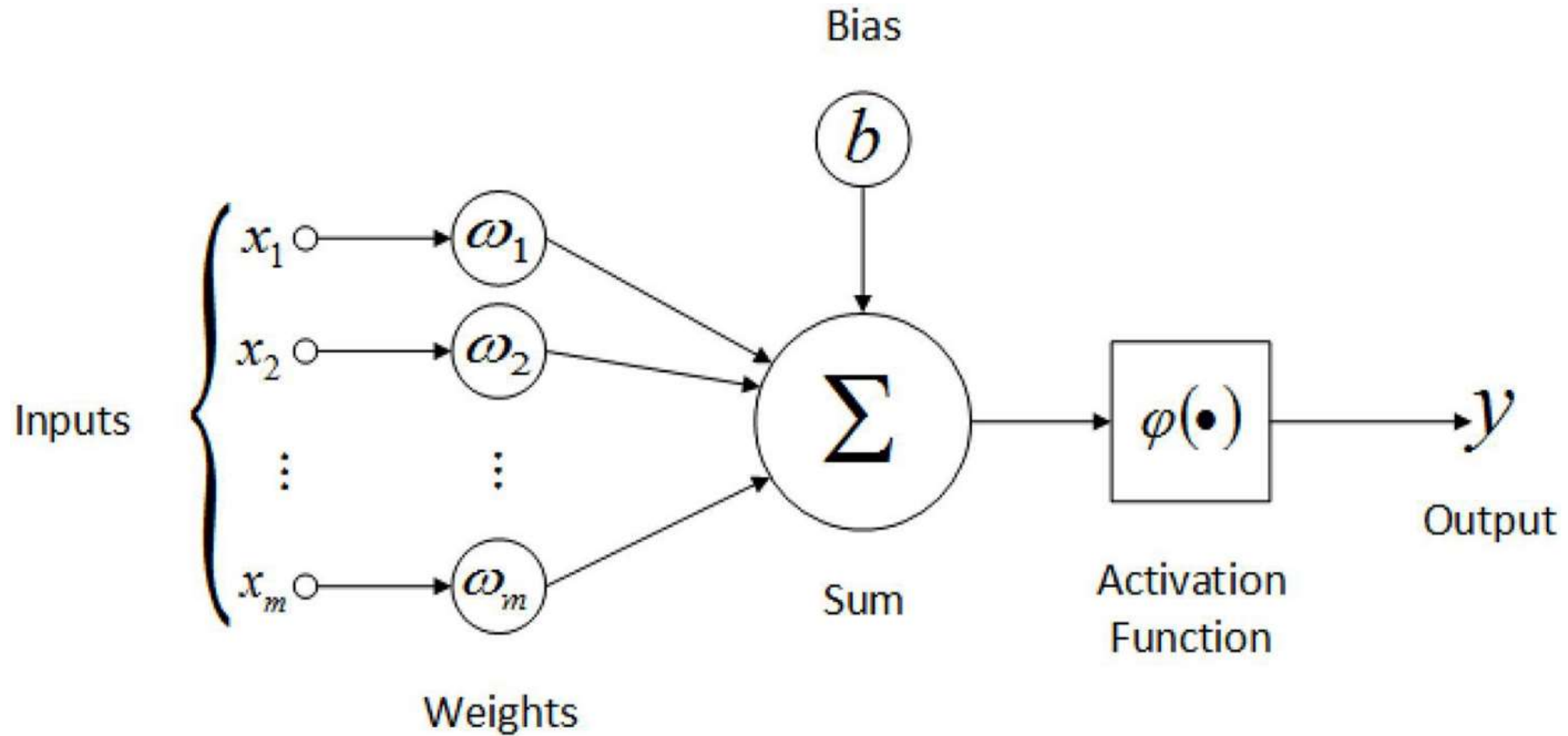
Conclusiones

Conclusiones

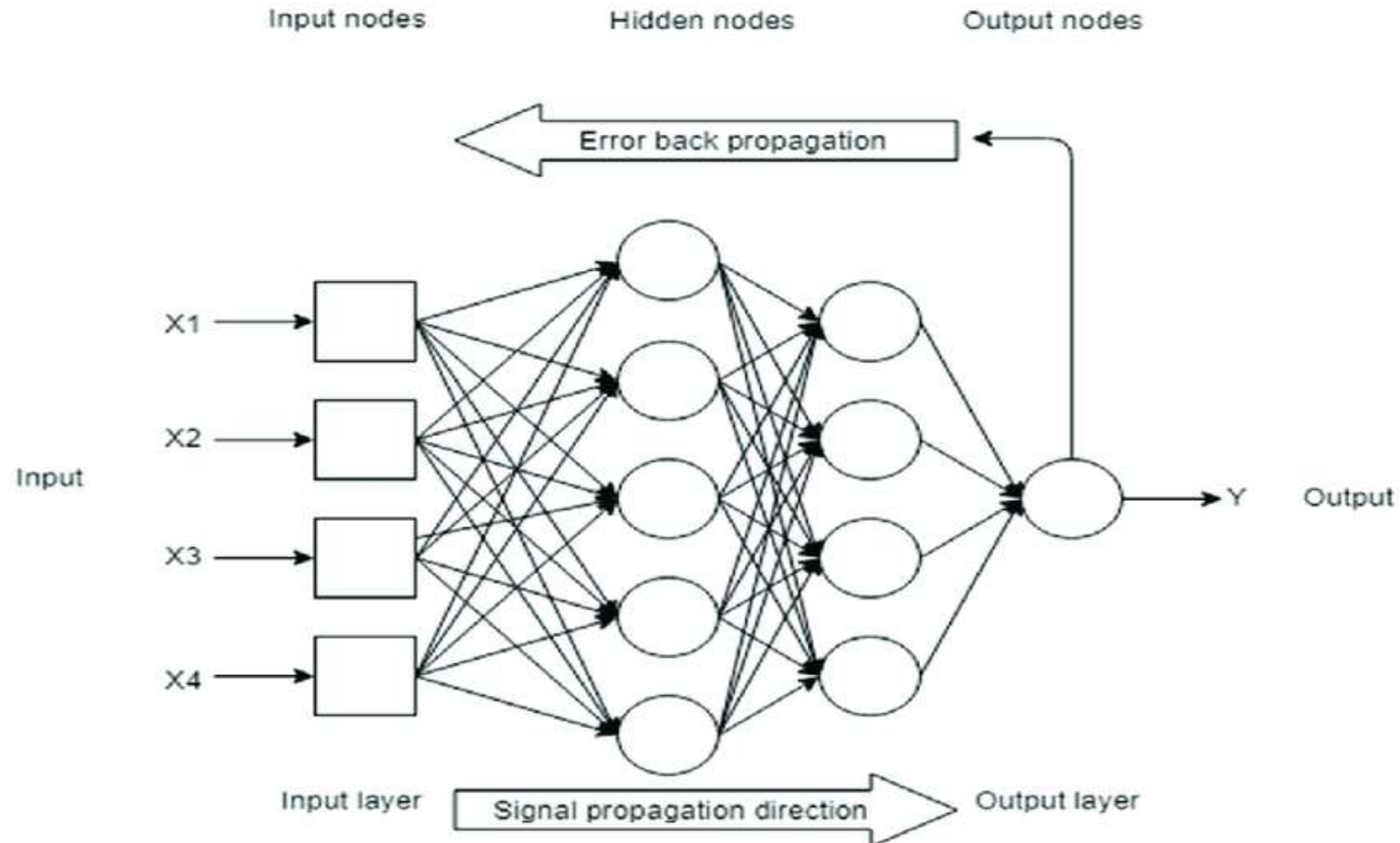
- Los modelos de WGAN son capaces de capturar los distintos sesgos y crear muestras realistas
- Distribuciones muy sesgadas favorecen el aprendizaje de la WGAN
- Los modelos han sido entrenados confinando los parámetros de D en $[-c, c]$
- La WGAN capta el sesgo en los datos y lo aplica al vector de información concatenado en el espacio latente

Muchas gracias por su atención

Funcionamiento de una neurona



Feedforward y backpropagation

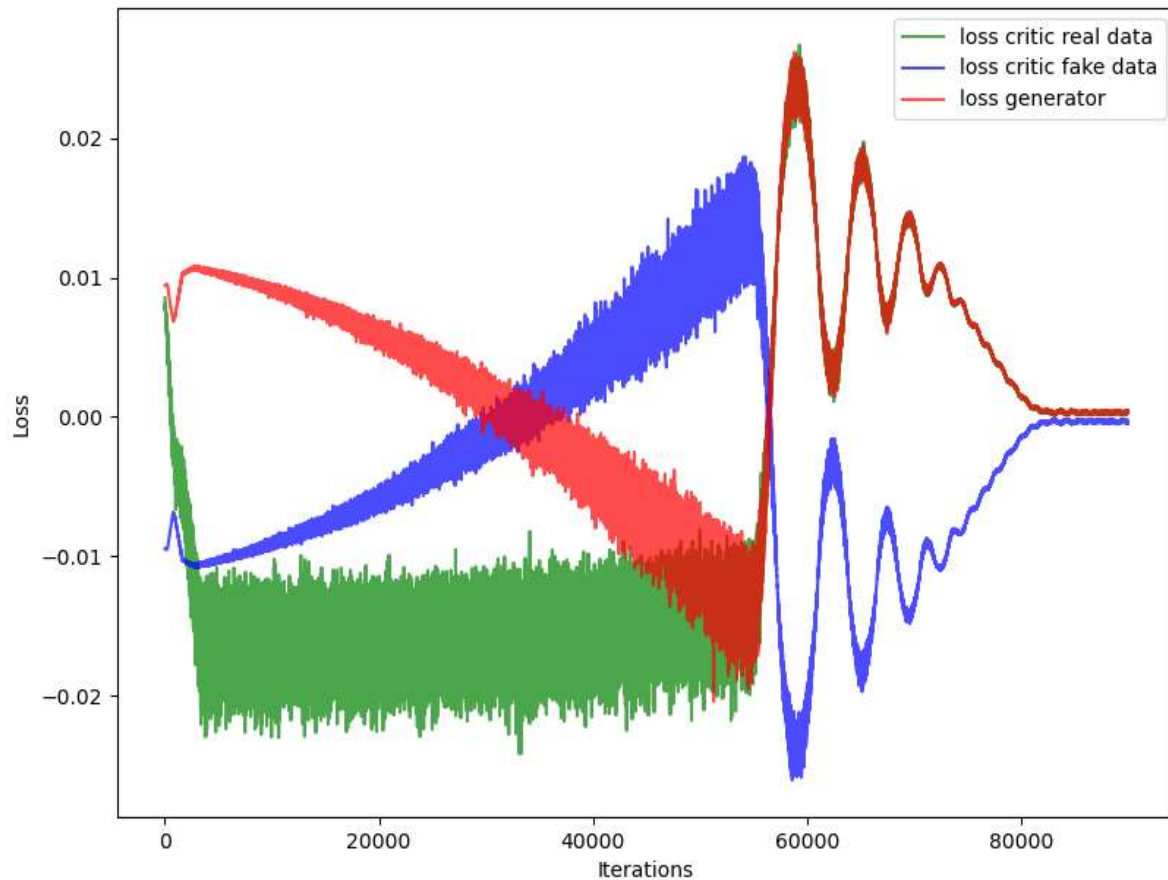


Función objetivo GAN

- Basado en:
 - Divergencia Jensen-Shannon
- Propósito de D :
 - Discriminar entre muestra real o falsa

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

Función de *Loss*

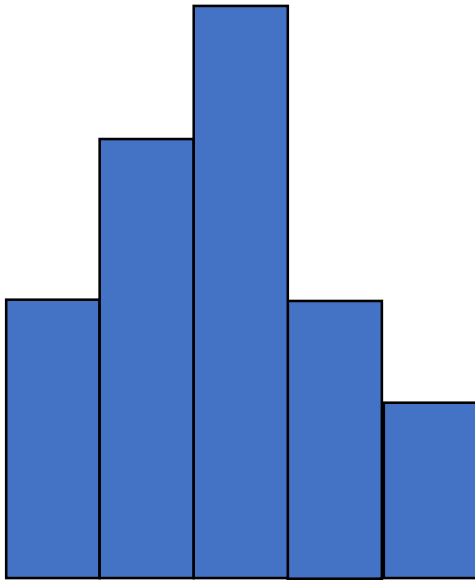


■ $Loss(D_{real}) = -\mathbb{E}_{x \sim p_{data}} [D(x)]$

■ $Loss(D_{fake}) = \mathbb{E}_{z \sim p_z} [D(G(z))]$

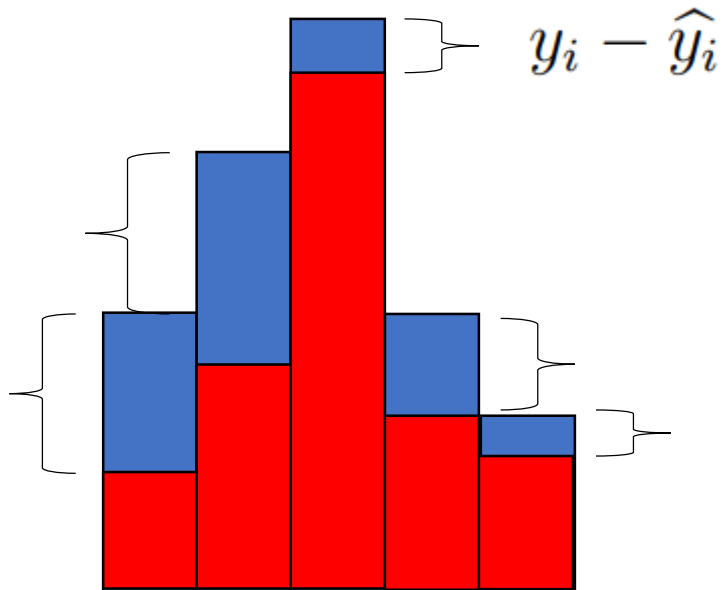
■ $Loss(G) = -\mathbb{E}_{z \sim p_z} [D(G(z))]$

Root-Mean Square Error



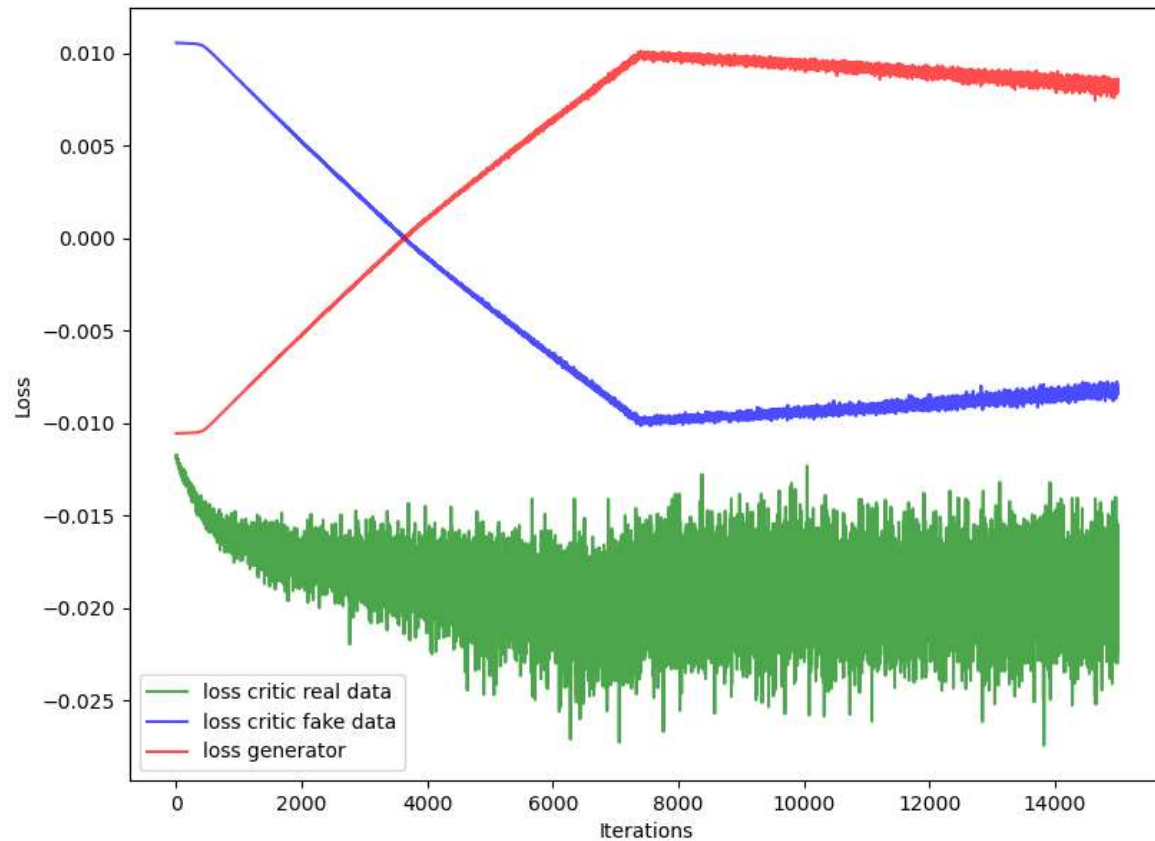
$$RMSE = \sqrt{\frac{1}{n} \sum_i^n (y_i - \hat{y}_i)^2}$$

Root-Mean Square Error

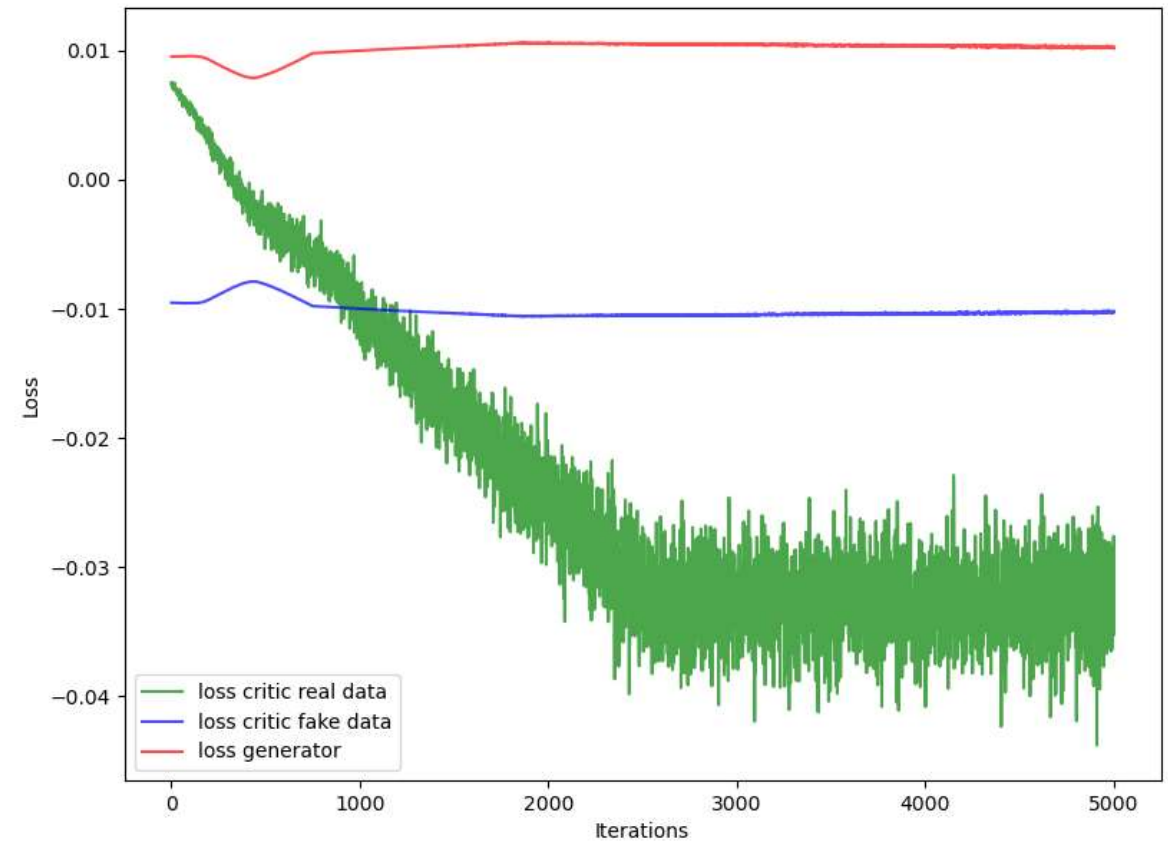


$$RMSE = \sqrt{\frac{1}{n} \sum_i^n (y_i - \hat{y}_i)^2}$$

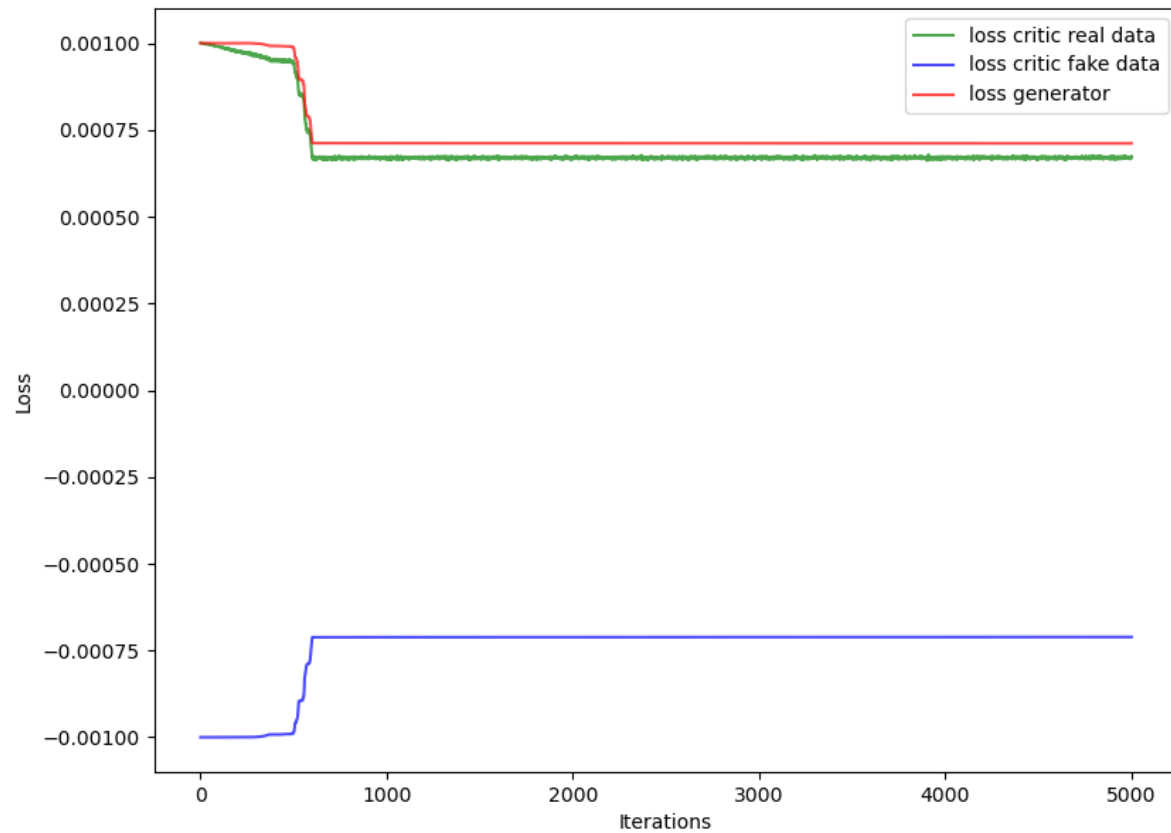
n-critic bajo



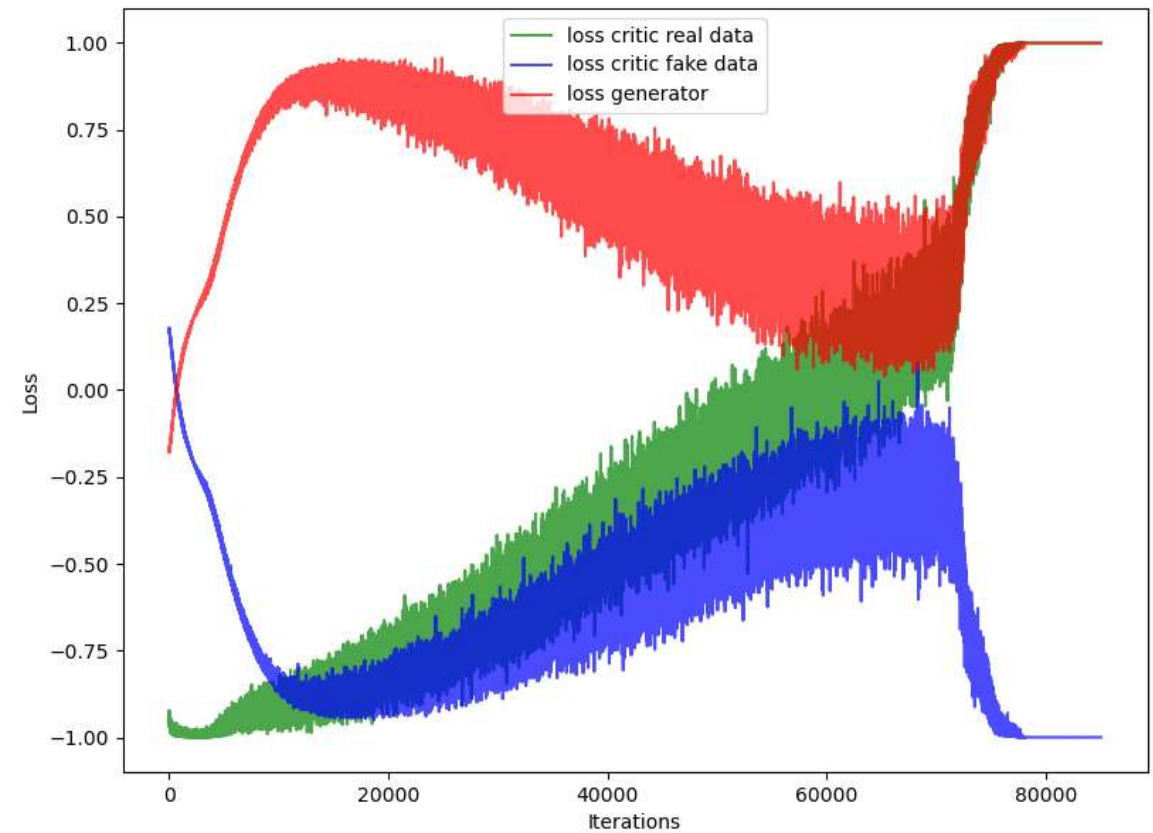
n-critic alto



c bajo

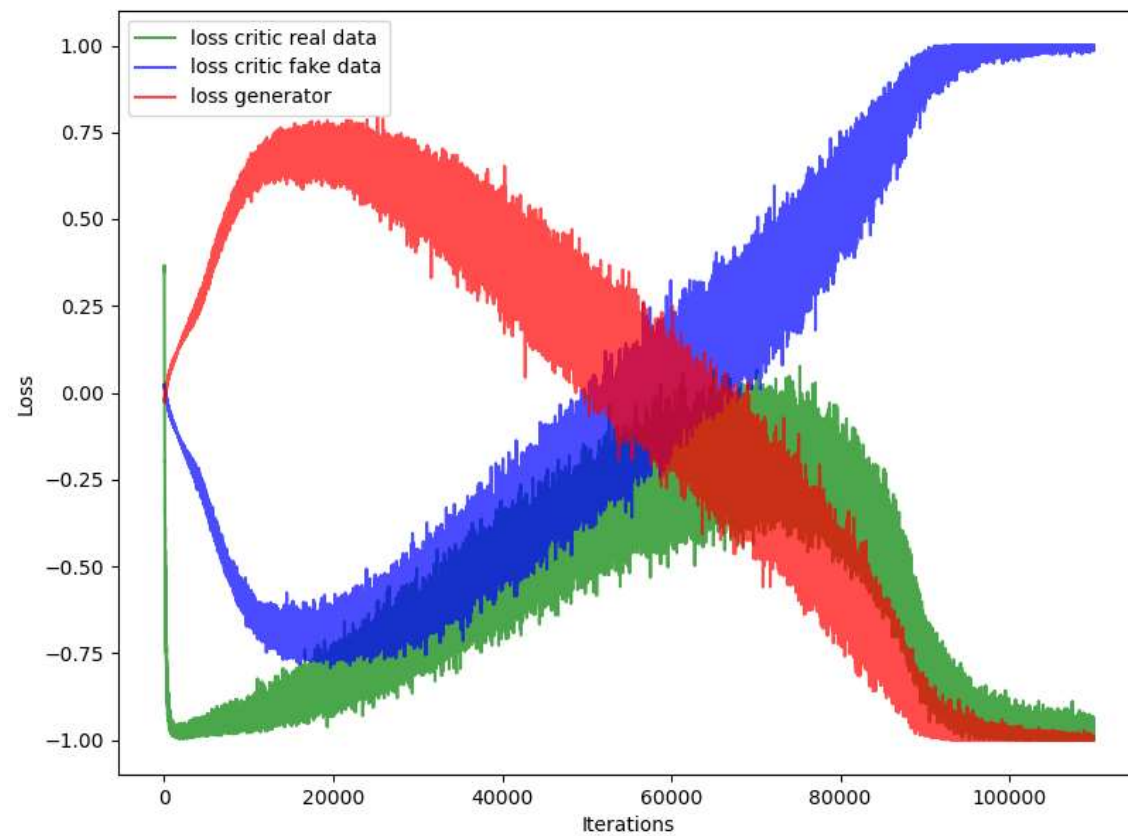
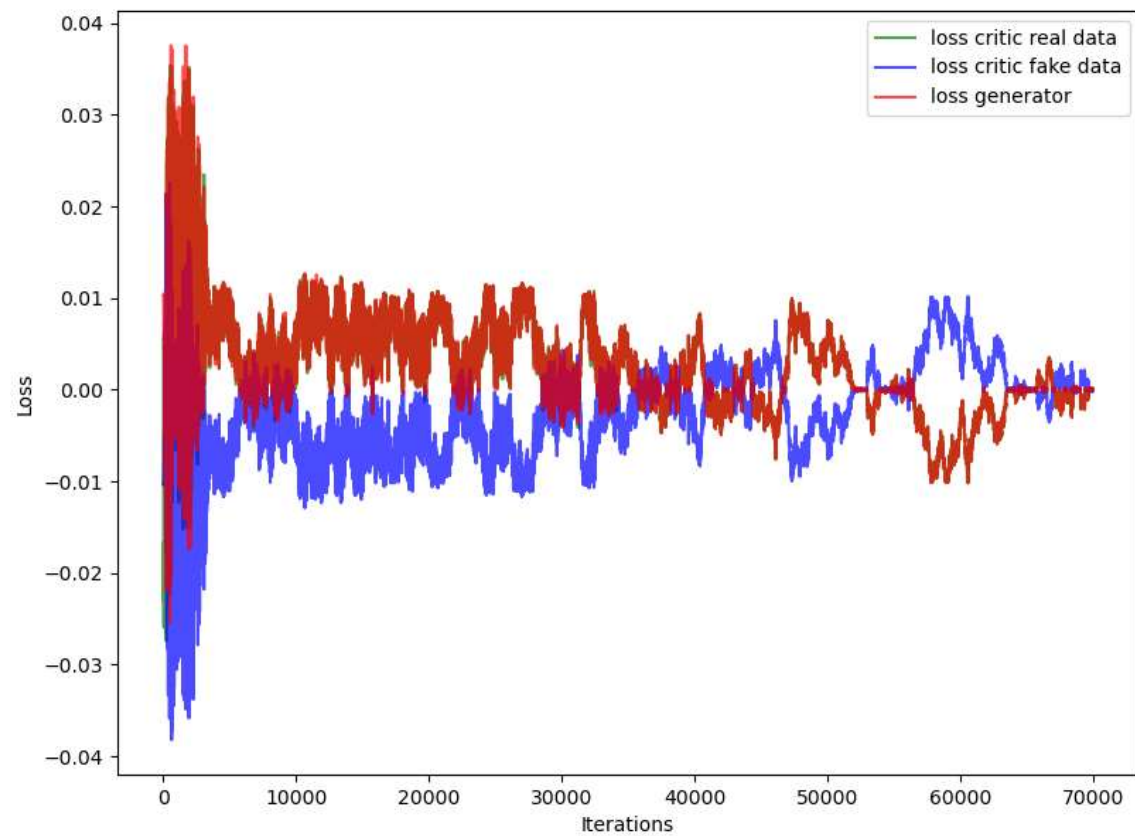


c alto

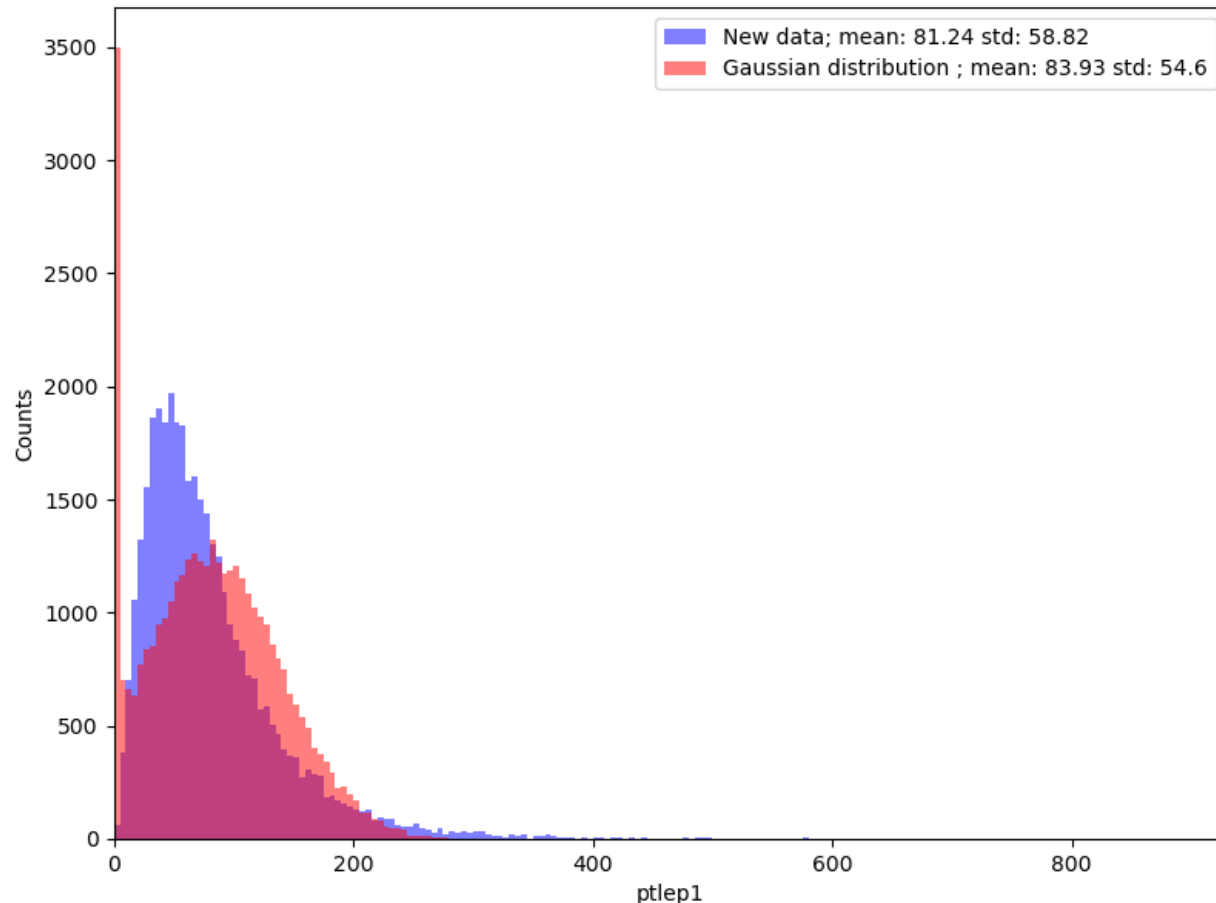


α alto

WGAN-GP $\lambda = 10$

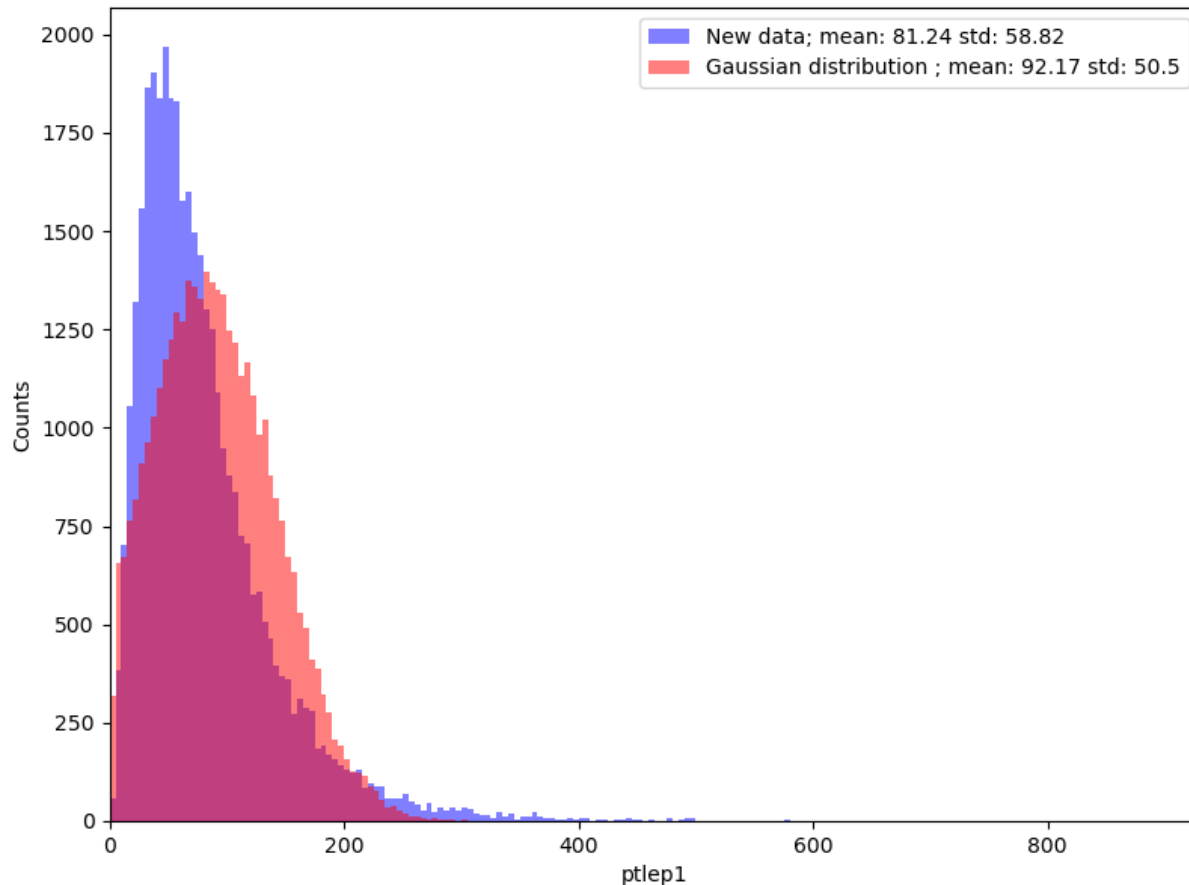


Capacidad de la WGAN



- Sesgo en la red del 60% en la media
- Espacio latente en el entrenamiento:
 - 10 canales de ruido uniforme
 - 1 canal de información (p_T)
- Espacio latente nuevo:
 - 10 canales de ruido uniforme
 - 1 canal de ruido gaussiano $N(\text{mean}=50.76, \text{std}=37.85)$
- Muestra resultante:
 - Distribución gaussiana
 - Valores negativos enviados al 0

Capacidad de la WGAN



- Sesgo en la red del 60% en la media
- Espacio latente en el entrenamiento:
 - 10 canales de ruido uniforme
 - 1 canal de información (p_T)
- Espacio latente nuevo:
 - 10 canales de ruido uniforme
 - 1 canal de ruido solo positivo gaussiano $N(\text{mean}=50.76, \text{std}=37.85)$
- Muestra resultante:
 - Distribución gaussiana sesgada