

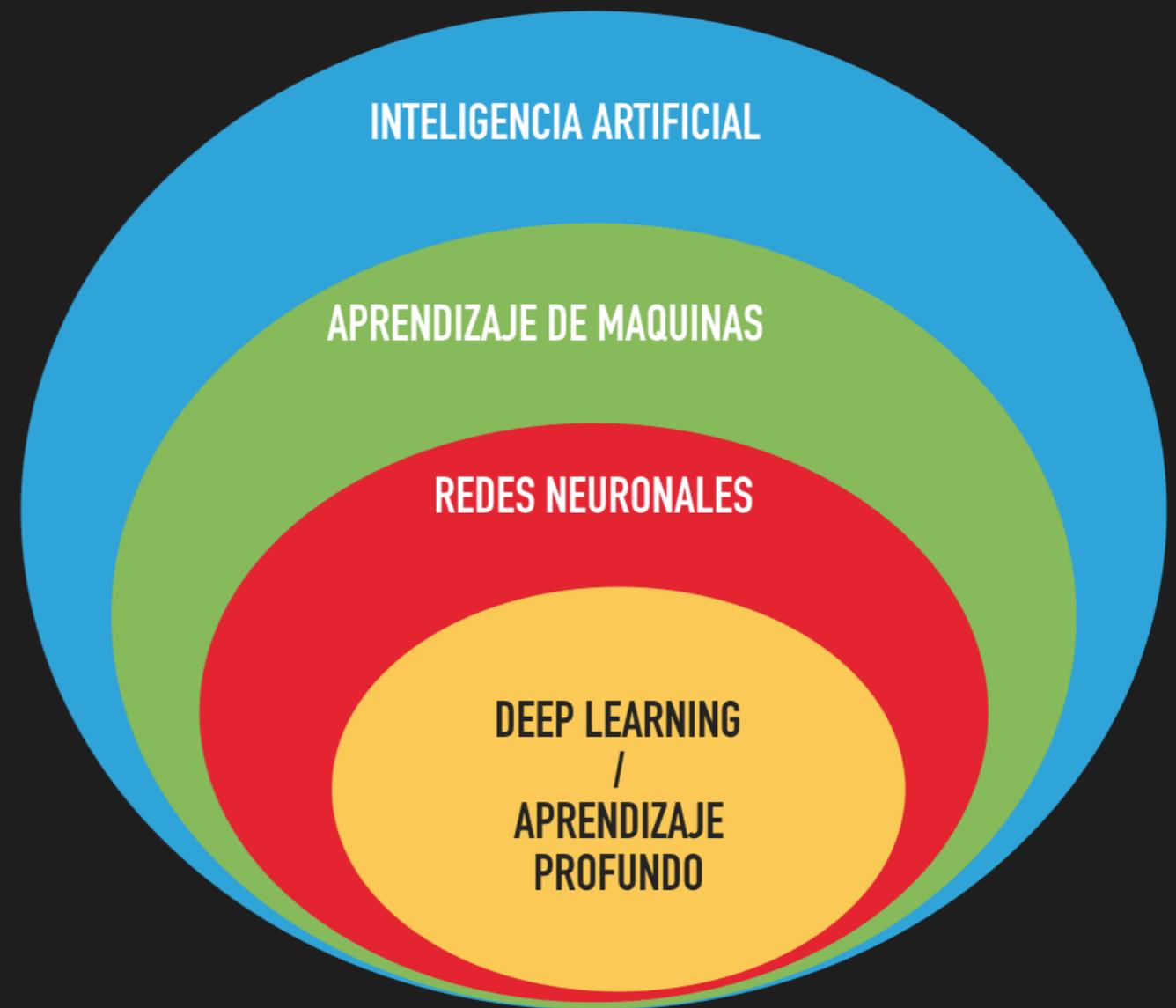
DIEGO GUTIERREZ CORONEL

IMPLEMENTACIONES DE ML EN FÍSICA DE PARTÍCULAS EXPERIMENTAL

quantil

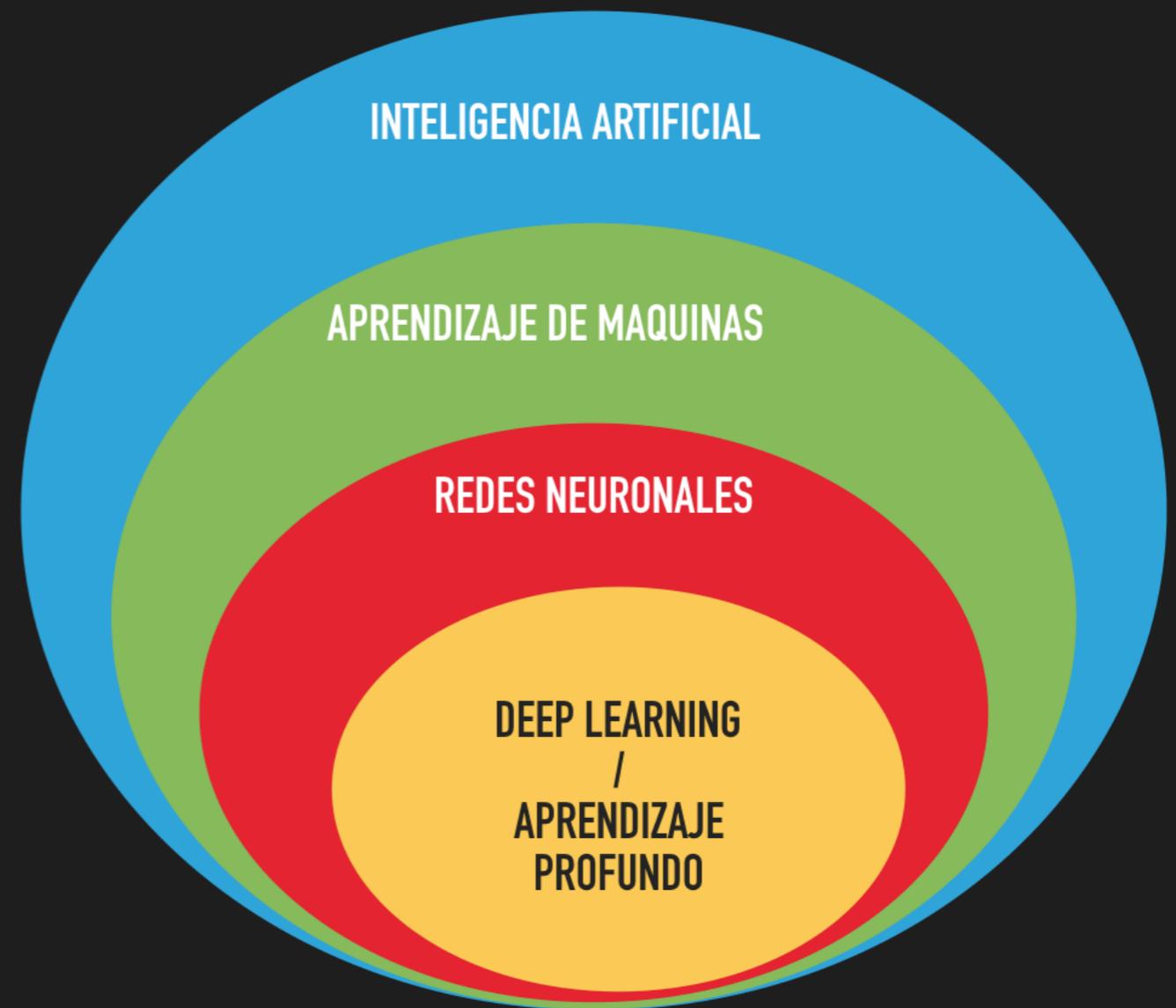
matemáticas aplicadas

MACHINE LEARNING



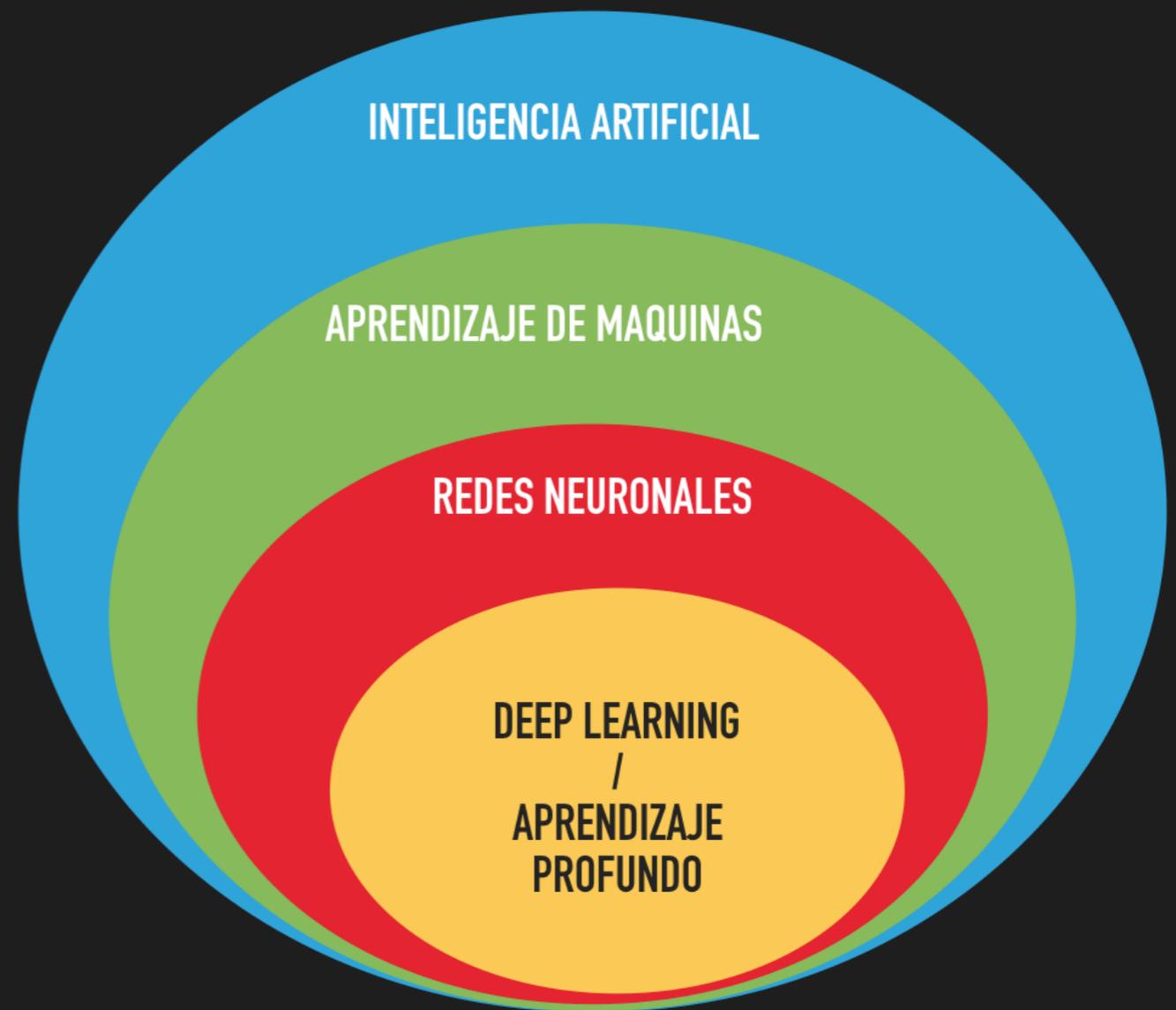
MACHINE LEARNING

- ▶ Aprendizaje de Maquinas \subset Inteligencia Artificial



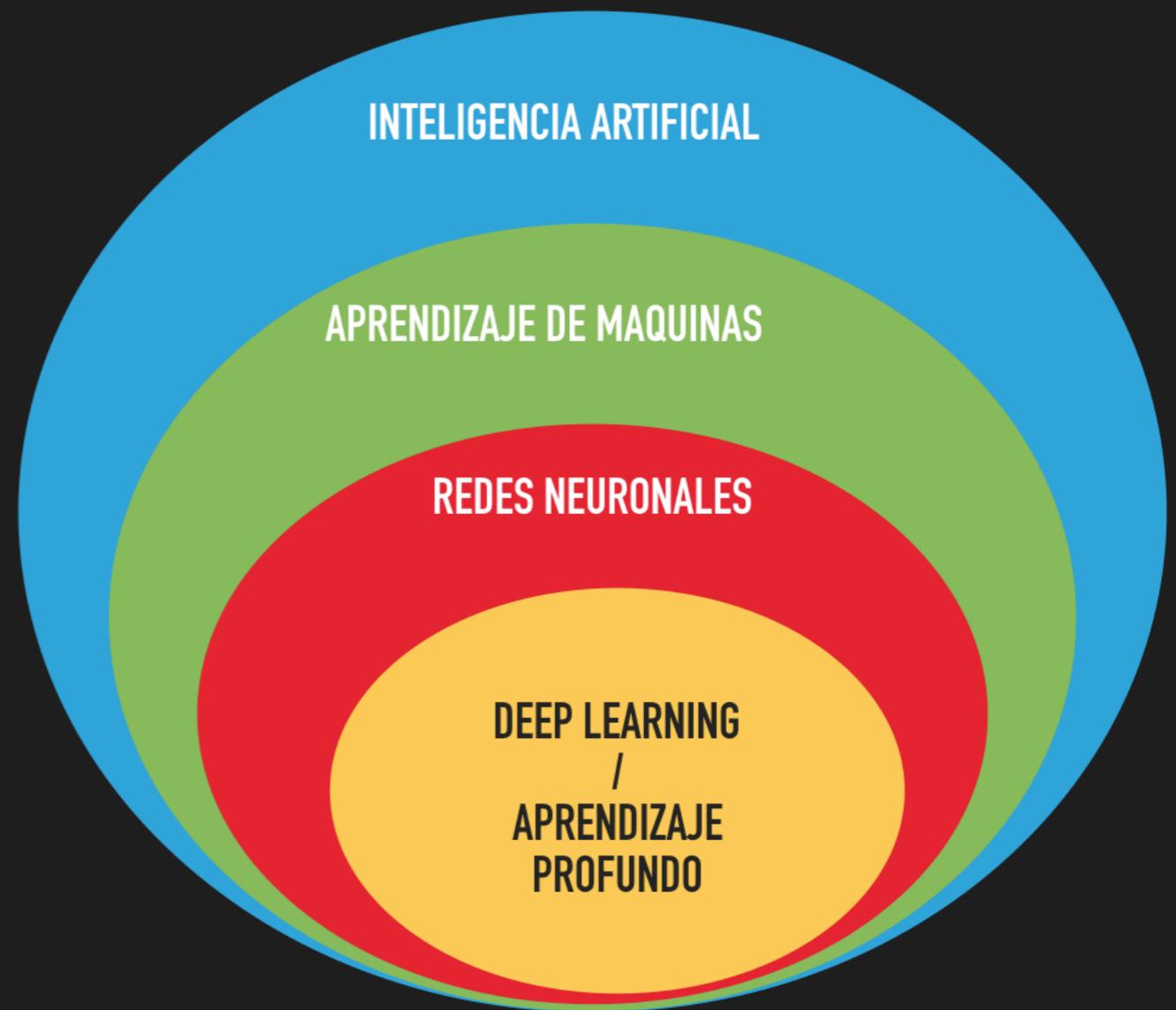
MACHINE LEARNING

- ▶ Aprendizaje de Maquinas \subset Inteligencia Artificial
- ▶ Método de Aprendizaje Estadístico usando datos (grandes cantidades)



MACHINE LEARNING

- ▶ Aprendizaje de Maquinas \subset Inteligencia Artificial
- ▶ Método de Aprendizaje Estadístico usando datos (grandes cantidades)
- ▶ Lenguaje



INTRO

FÍSICA DE PARTÍCULAS

FÍSICA DE PARTÍCULAS

- ▶ Estudia los componentes fundamentales de la materia y sus interacciones

FÍSICA DE PARTÍCULAS

- ▶ Estudia los componentes fundamentales de la materia y sus interacciones
- ▶ Modelo Estándar

FÍSICA DE PARTÍCULAS

- ▶ Estudia los componentes fundamentales de la materia y sus interacciones
- ▶ Modelo Estándar
- ▶ Una enorme cantidad de misterios por resolver y un reto para la humanidad

FÍSICA DE PARTÍCULAS

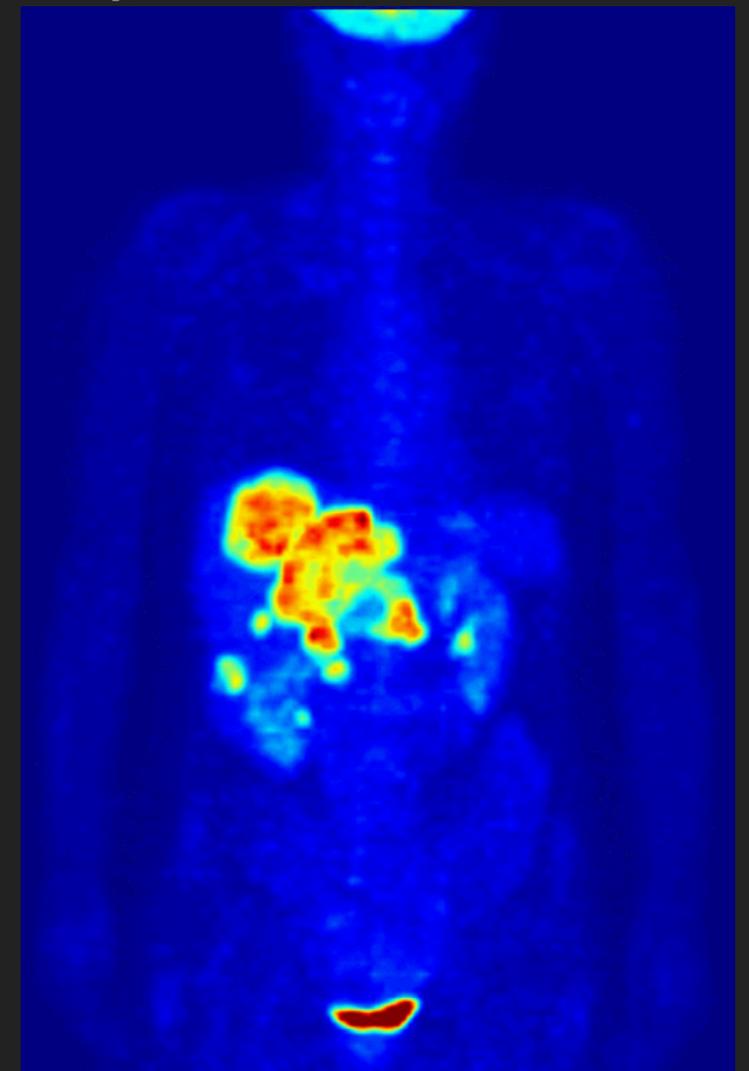
- ▶ Estudia los componentes fundamentales de la materia y sus interacciones
- ▶ Modelo Estándar
- ▶ Una enorme cantidad de misterios por resolver y un reto para la humanidad
- ▶ Fuente de tecnologías usadas al día de hoy

FÍSICA DE PARTÍCULAS

- ▶ Estudia los componentes fundamentales de la materia y sus interacciones
- ▶ Modelo Estándar
- ▶ Una enorme cantidad de misterios por resolver y un reto para la humanidad
- ▶ Fuente de tecnologías usadas al día de hoy
 - ▶ Computo

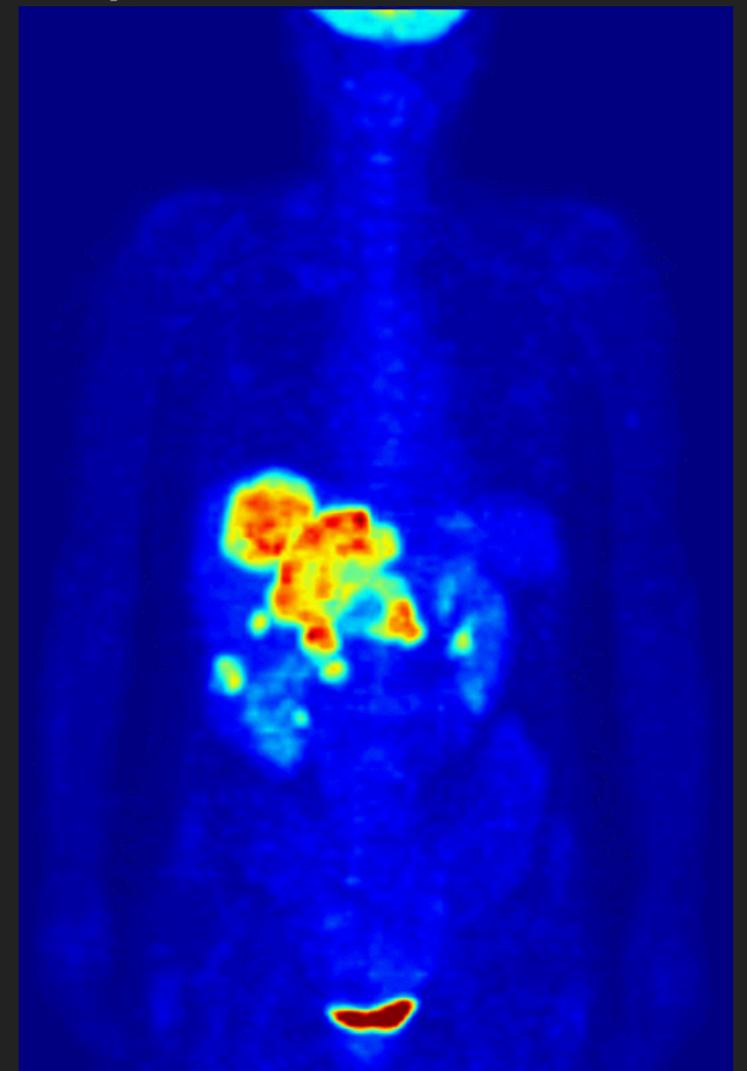
FÍSICA DE PARTÍCULAS

- ▶ Estudia los componentes fundamentales de la materia y sus interacciones
- ▶ Modelo Estándar
- ▶ Una enorme cantidad de misterios por resolver y un reto para la humanidad
- ▶ Fuente de tecnologías usadas al día de hoy
 - ▶ Computo



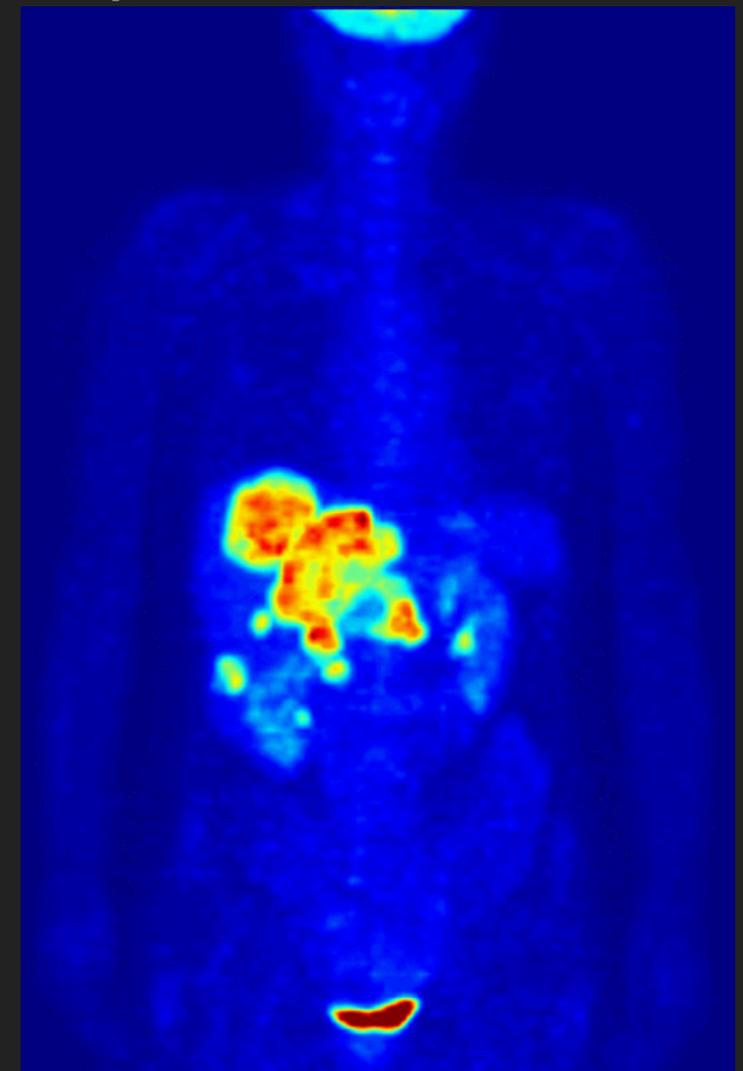
FÍSICA DE PARTÍCULAS

- ▶ Estudia los componentes fundamentales de la materia y sus interacciones
- ▶ Modelo Estándar
- ▶ Una enorme cantidad de misterios por resolver y un reto para la humanidad
- ▶ Fuente de tecnologías usadas al día de hoy
 - ▶ Computo
 - ▶ Detectores



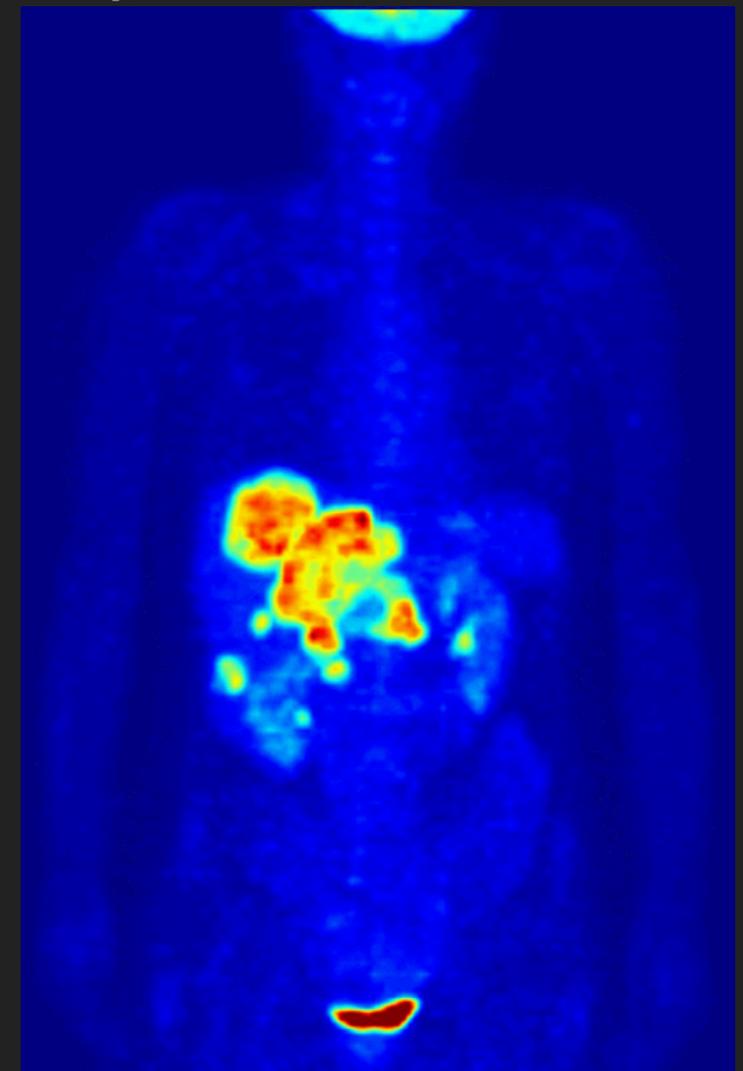
FÍSICA DE PARTÍCULAS

- ▶ Estudia los componentes fundamentales de la materia y sus interacciones
- ▶ Modelo Estándar
- ▶ Una enorme cantidad de misterios por resolver y un reto para la humanidad
- ▶ Fuente de tecnologías usadas al día de hoy
 - ▶ Computo
 - ▶ Detectores
 - ▶ Aceleradores



FÍSICA DE PARTÍCULAS

- ▶ Estudia los componentes fundamentales de la materia y sus interacciones
- ▶ Modelo Estándar
- ▶ Una enorme cantidad de misterios por resolver y un reto para la humanidad
- ▶ Fuente de tecnologías usadas al día de hoy
 - ▶ Computo
 - ▶ Detectores
 - ▶ Aceleradores



INTRO

OUTLINE

OUTLINE

- Introducción a Física de Partículas

OUTLINE

- Introducción a Física de Partículas
- Gran Colisionador de Hadrones y el Big Data

OUTLINE

- Introducción a Física de Partículas
- Gran Colisionador de Hadrones y el Big Data
 - Detectores

OUTLINE

- Introducción a Física de Partículas
- Gran Colisionador de Hadrones y el Big Data
 - Detectores
- Aprendizaje de Maquinas y sus aplicaciones

OUTLINE

- Introducción a Física de Partículas
- Gran Colisionador de Hadrones y el Big Data
 - Detectores
- Aprendizaje de Maquinas y sus aplicaciones
 - Aprendizaje supervisado, no-supervisado, aprendizaje por refuerzo

OUTLINE

- Introducción a Física de Partículas
- Gran Colisionador de Hadrones y el Big Data
 - Detectores
- Aprendizaje de Maquinas y sus aplicaciones
 - Aprendizaje supervisado, no-supervisado, aprendizaje por refuerzo
- Algunas Implementaciones a Física de Partículas

OUTLINE

- Introducción a Física de Partículas
- Gran Colisionador de Hadrones y el Big Data
 - Detectores
- Aprendizaje de Maquinas y sus aplicaciones
 - Aprendizaje supervisado, no-supervisado, aprendizaje por refuerzo
- Algunas Implementaciones a Física de Partículas
 - Diferentes Modelos

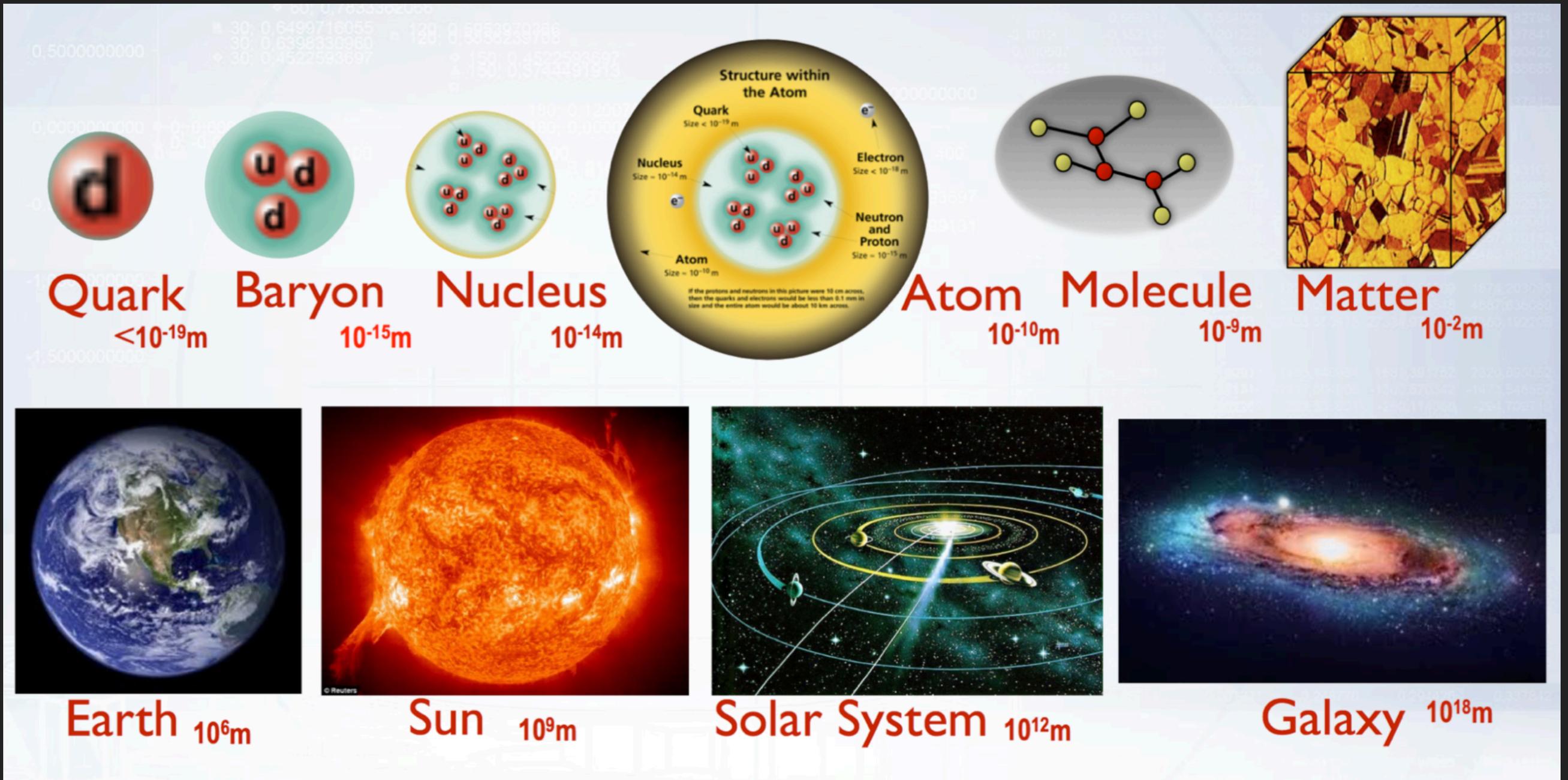
OUTLINE

- Introducción a Física de Partículas
- Gran Colisionador de Hadrones y el Big Data
 - Detectores
- Aprendizaje de Maquinas y sus aplicaciones
 - Aprendizaje supervisado, no-supervisado, aprendizaje por refuerzo
- Algunas Implementaciones a Física de Partículas
 - Diferentes Modelos
 - Redes Neuronales (Deep Learning)

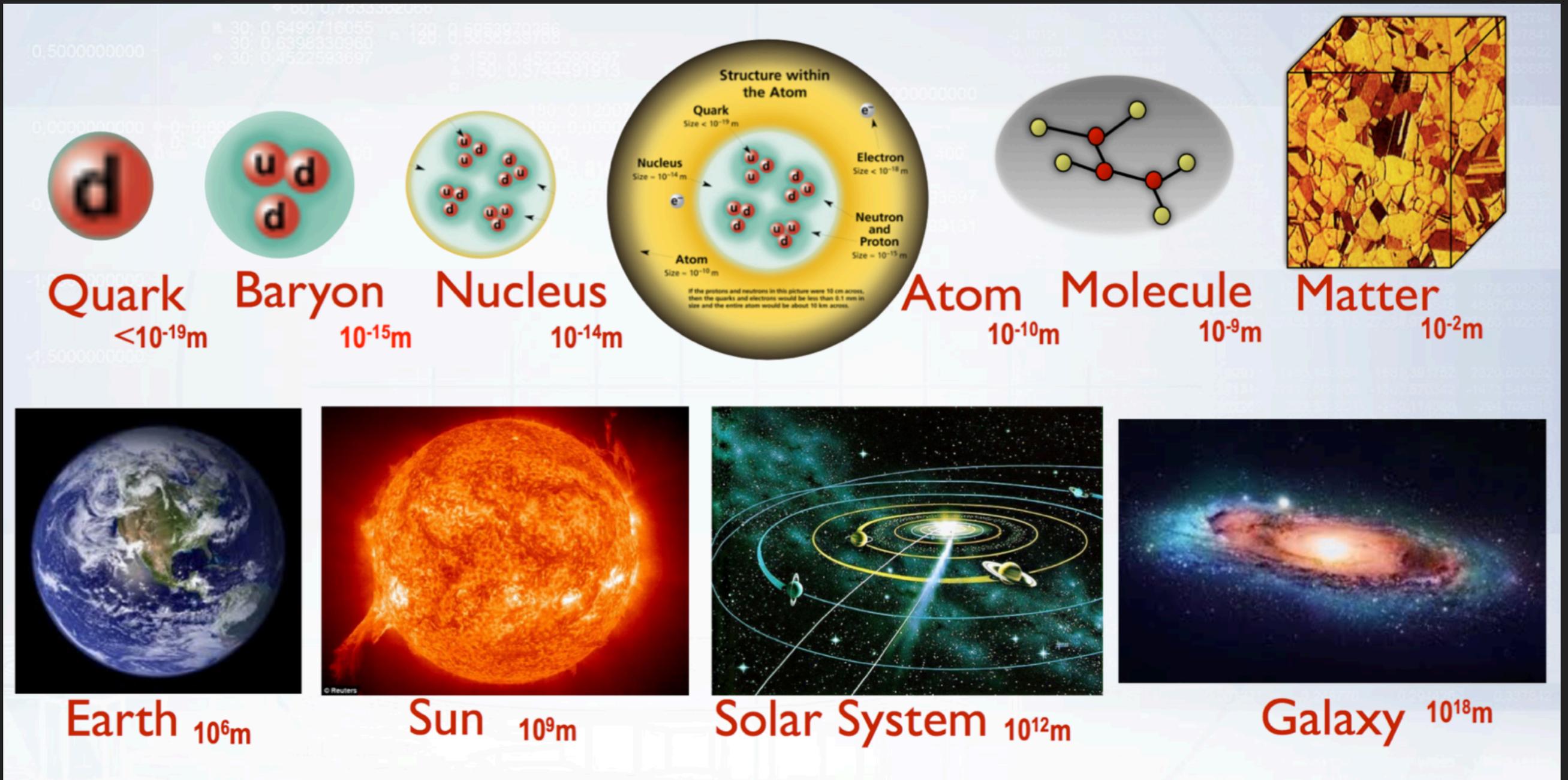
OUTLINE

- Introducción a Física de Partículas
- Gran Colisionador de Hadrones y el Big Data
 - Detectores
- Aprendizaje de Maquinas y sus aplicaciones
 - Aprendizaje supervisado, no-supervisado, aprendizaje por refuerzo
- Algunas Implementaciones a Física de Partículas
 - Diferentes Modelos
 - Redes Neuronales (Deep Learning)
- Más implementaciones

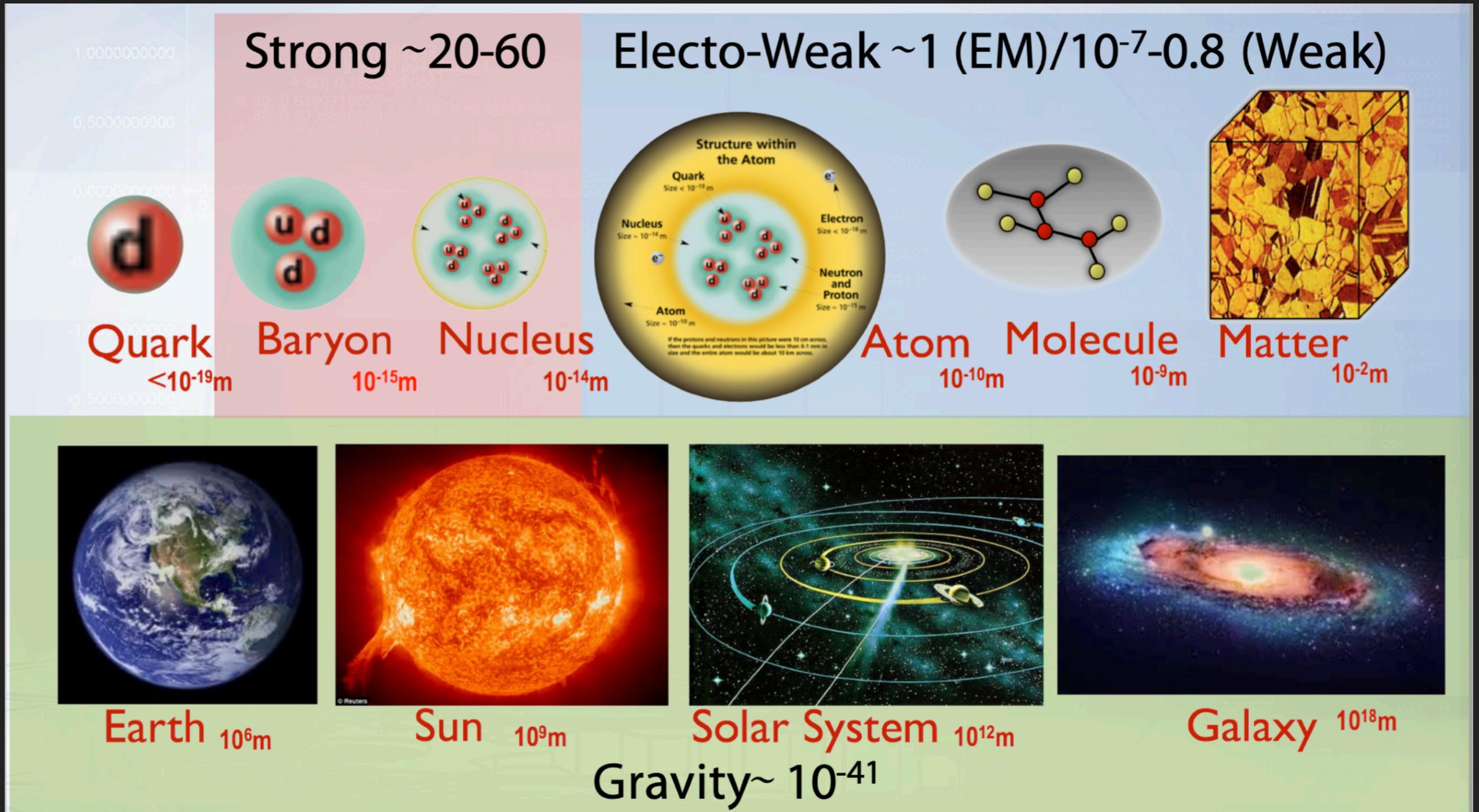
ESCALA DEL UNIVERSO



ESCALA DEL UNIVERSO



ESCALA DEL UNIVERSO



FÍSICA DE PARTÍCULAS

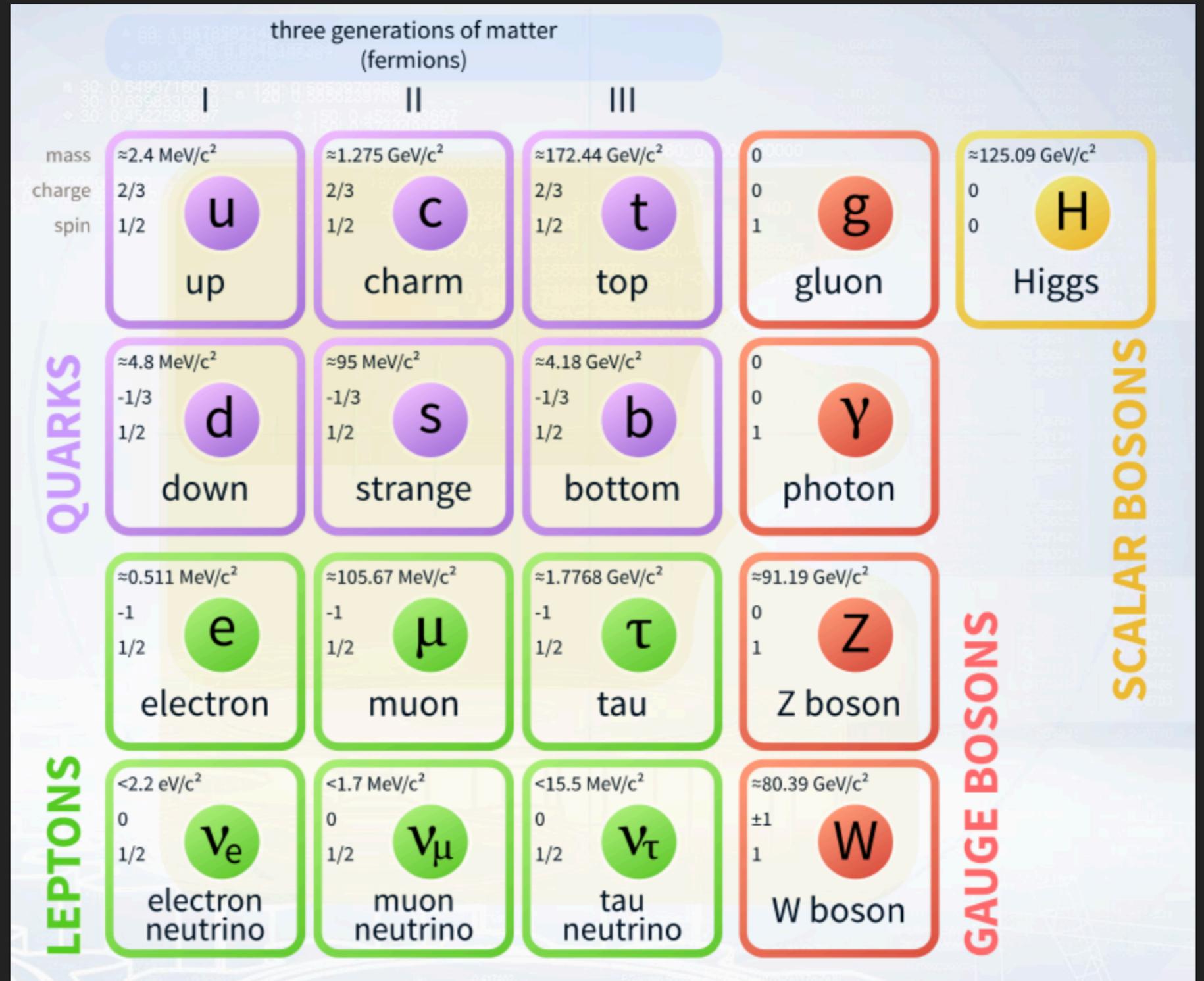
MODELO ESTÁNDAR

MODELO ESTÁNDAR

- ▶ Es una teoría física que describe las partículas fundamentales y su interacción

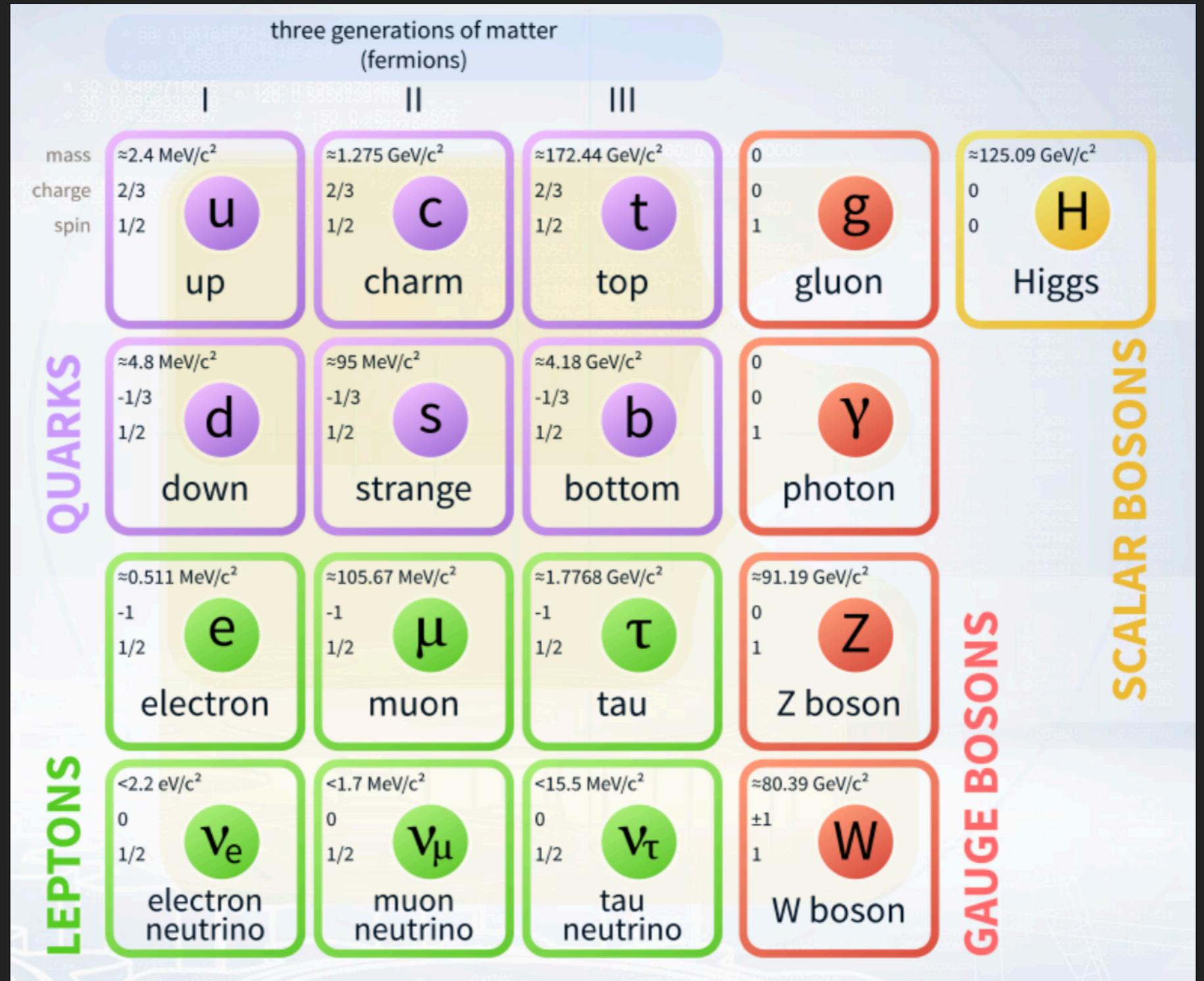
MODELO ESTÁNDAR

- ▶ Es una teoría física que describe las partículas fundamentales y su interacción



MODELO ESTÁNDAR

- ▶ Es una teoría física que describe las partículas fundamentales y su interacción
- ▶ Fermiones - spin semi-entero ($1/2, 3/2, 5/2, \dots$)



MODELO ESTÁNDAR

- ▶ Es una teoría física que describe las partículas fundamentales y su interacción
- ▶ Fermiones - spin semi-entero ($1/2, 3/2, 5/2, \dots$)
- ▶ Bosones - spin entero ($1, 2, 3, \dots$)

three generations of matter (fermions)

	I	II	III		
mass	$\approx 2.4 \text{ MeV}/c^2$	$\approx 1.275 \text{ GeV}/c^2$	$\approx 172.44 \text{ GeV}/c^2$	0	$\approx 125.09 \text{ GeV}/c^2$
charge	$2/3$	$2/3$	$2/3$	0	0
spin	$1/2$	$1/2$	$1/2$	1	0
	u up	c charm	t top	g gluon	H Higgs
	$\approx 4.8 \text{ MeV}/c^2$	$\approx 95 \text{ MeV}/c^2$	$\approx 4.18 \text{ GeV}/c^2$	0	
	$-1/3$	$-1/3$	$-1/3$	0	
	$1/2$	$1/2$	$1/2$	1	
	d down	s strange	b bottom	γ photon	
	$\approx 0.511 \text{ MeV}/c^2$	$\approx 105.67 \text{ MeV}/c^2$	$\approx 1.7768 \text{ GeV}/c^2$	$\approx 91.19 \text{ GeV}/c^2$	
	-1	-1	-1	0	
	$1/2$	$1/2$	$1/2$	1	
	e electron	μ muon	τ tau	Z Z boson	
	$< 2.2 \text{ eV}/c^2$	$< 1.7 \text{ MeV}/c^2$	$< 15.5 \text{ MeV}/c^2$	$\approx 80.39 \text{ GeV}/c^2$	
	0	0	0	± 1	
	$1/2$	$1/2$	$1/2$	1	
	ν_e electron neutrino	ν_μ muon neutrino	ν_τ tau neutrino	W W boson	

QUARKS

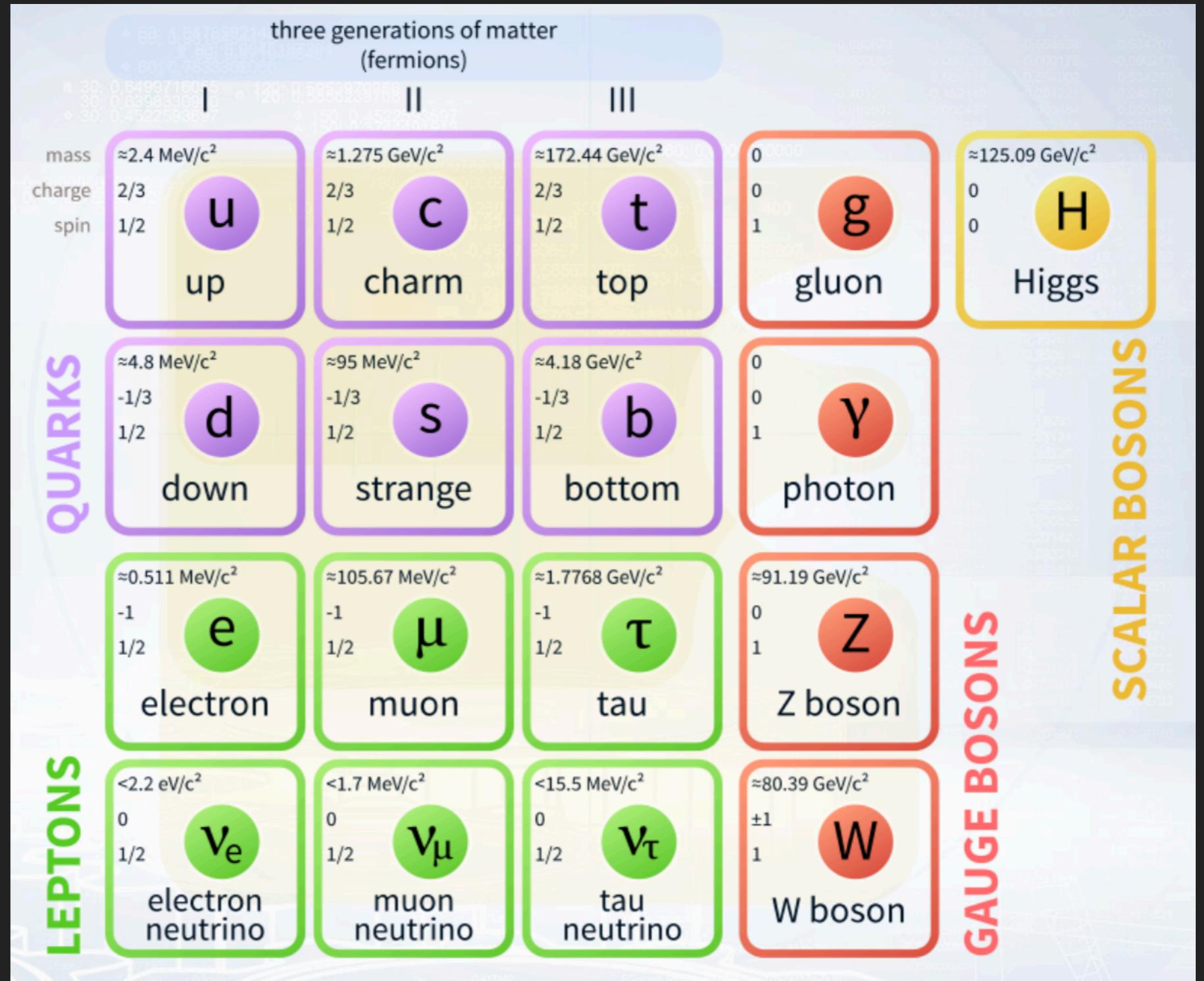
LEPTONS

GAUGE BOSONS

SCALAR BOSONS

MODELO ESTÁNDAR

- ▶ Es una teoría física que describe las partículas fundamentales y su interacción
- ▶ Fermiones - spin semi-entero ($1/2, 3/2, 5/2, \dots$)
- ▶ Bosones - spin entero ($1, 2, 3, \dots$)
- ▶ 6 quarks, 6 leptones, 4 bosons spin=1, 1 boson spin=0

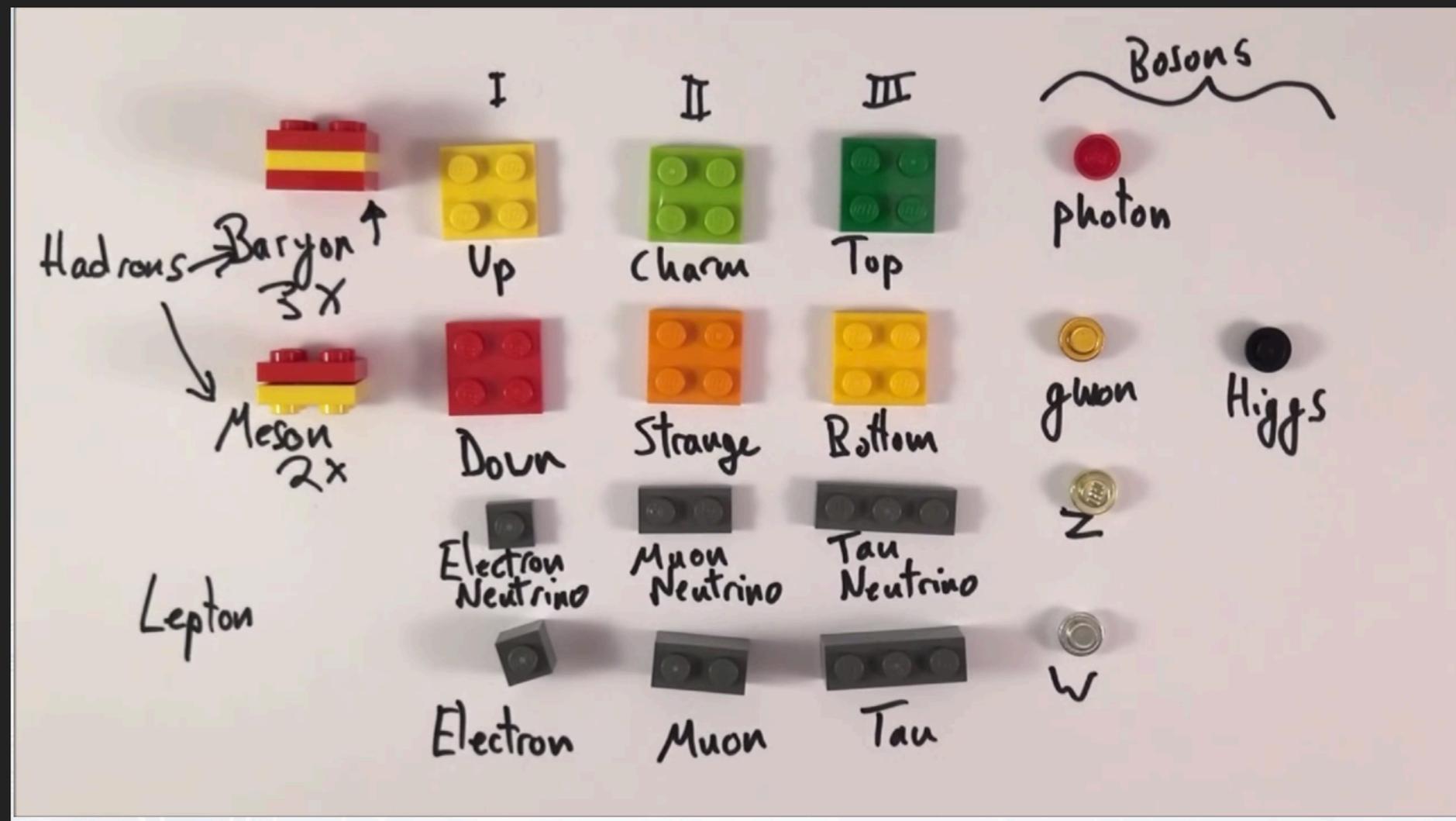


MODELO ESTÁNDAR

- ▶ Es una teoría física que describe las partículas fundamentales y su interacción
- ▶ Fermiones - spin semi-entero ($1/2, 3/2, 5/2, \dots$)
- ▶ Bosones - spin entero ($1, 2, 3, \dots$)
- ▶ 6 quarks, 6 leptones, 4 bosons spin=1, 1 boson spin=0

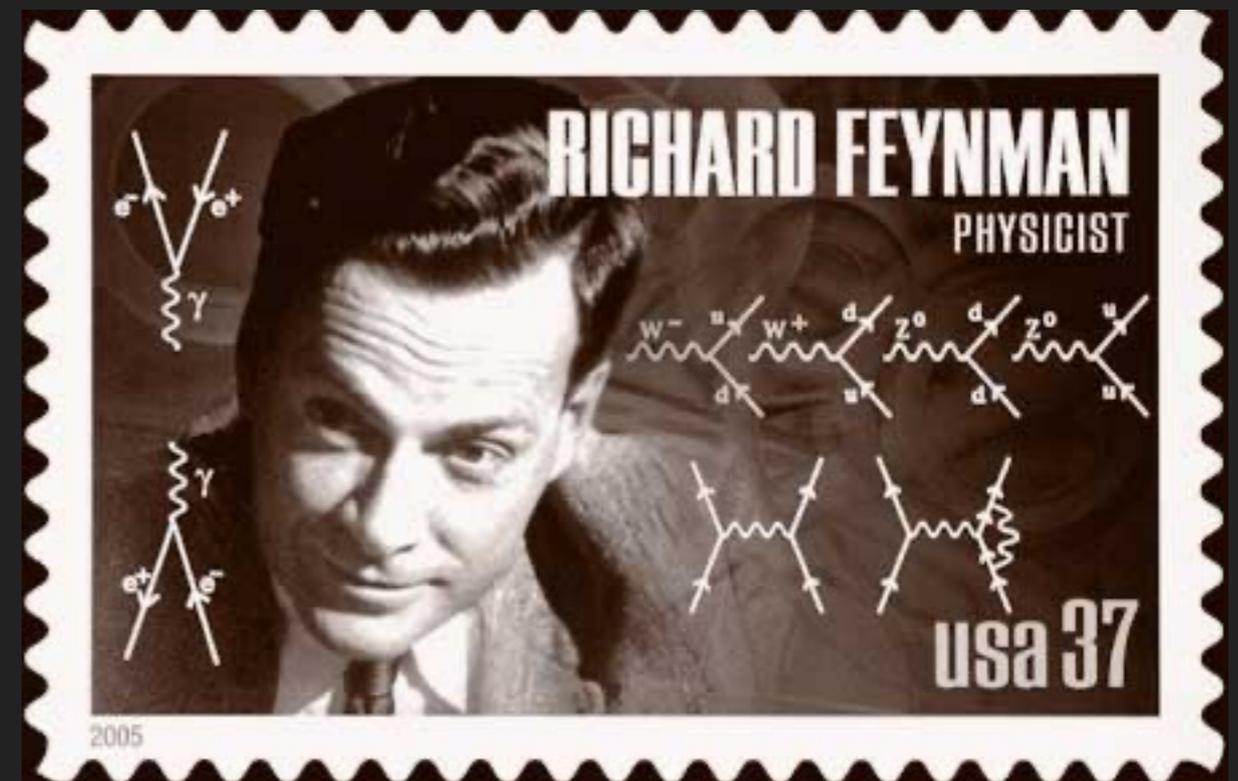
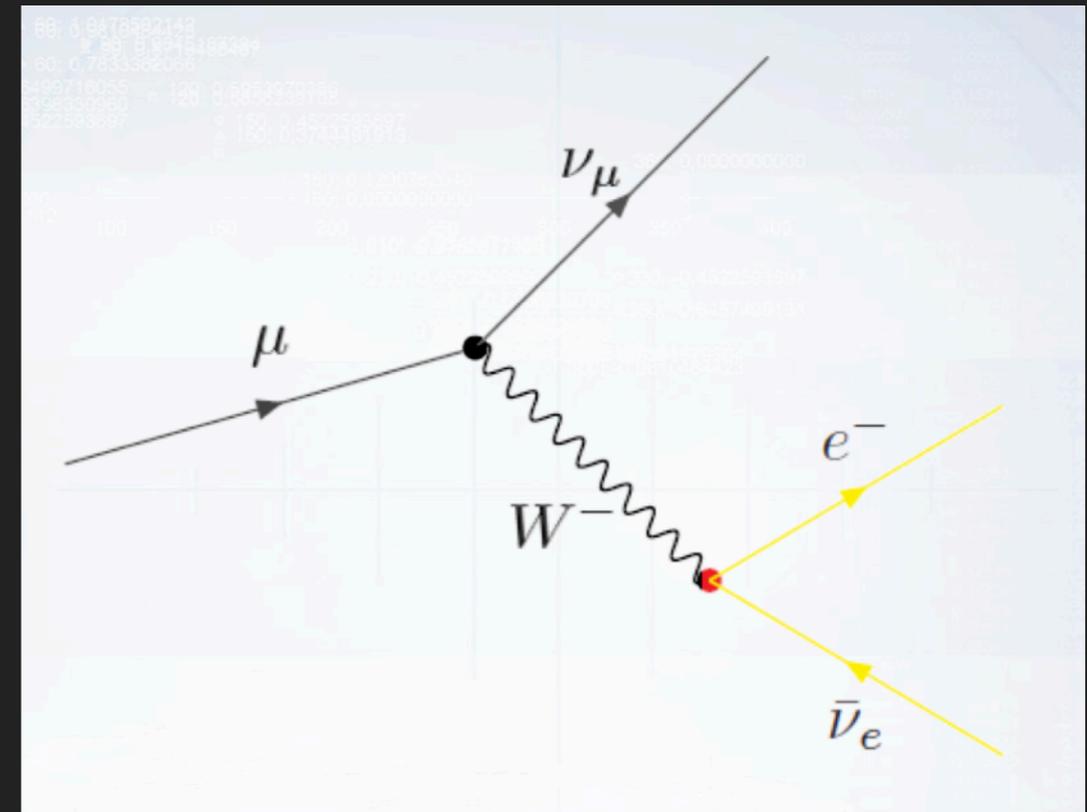
MODELO ESTÁNDAR

- ▶ Es una teoría física que describe las partículas fundamentales y su interacción
- ▶ Fermiones - spin semi-entero ($1/2, 3/2, 5/2, \dots$)
- ▶ Bosones - spin entero ($1, 2, 3, \dots$)
- ▶ 6 quarks, 6 leptones, 4 bosons spin=1, 1 boson spin=0



DIAGRAMAS DE FEYNMANN

- ▶ El lenguaje físico-matemático que se usa es la Teoría Cuántica de Campos (QFT)
- ▶ Partículas y bosones
- ▶ Permiten calcular probabilidades de procesos y tasa de desintegración (decay rate)



LIMITACIONES MODELO ESTÁNDAR Y FÍSICA MÁS ALLÁ

- ▶ Asimetría entre la cantidad de materia y antimateria en el universo
- ▶ Expansión acelerada del universo
- ▶ Gravedad?
- ▶ Materia oscura -> SUSY - Teoría de Supersimetría
- ▶ Oscilación de neutrinos

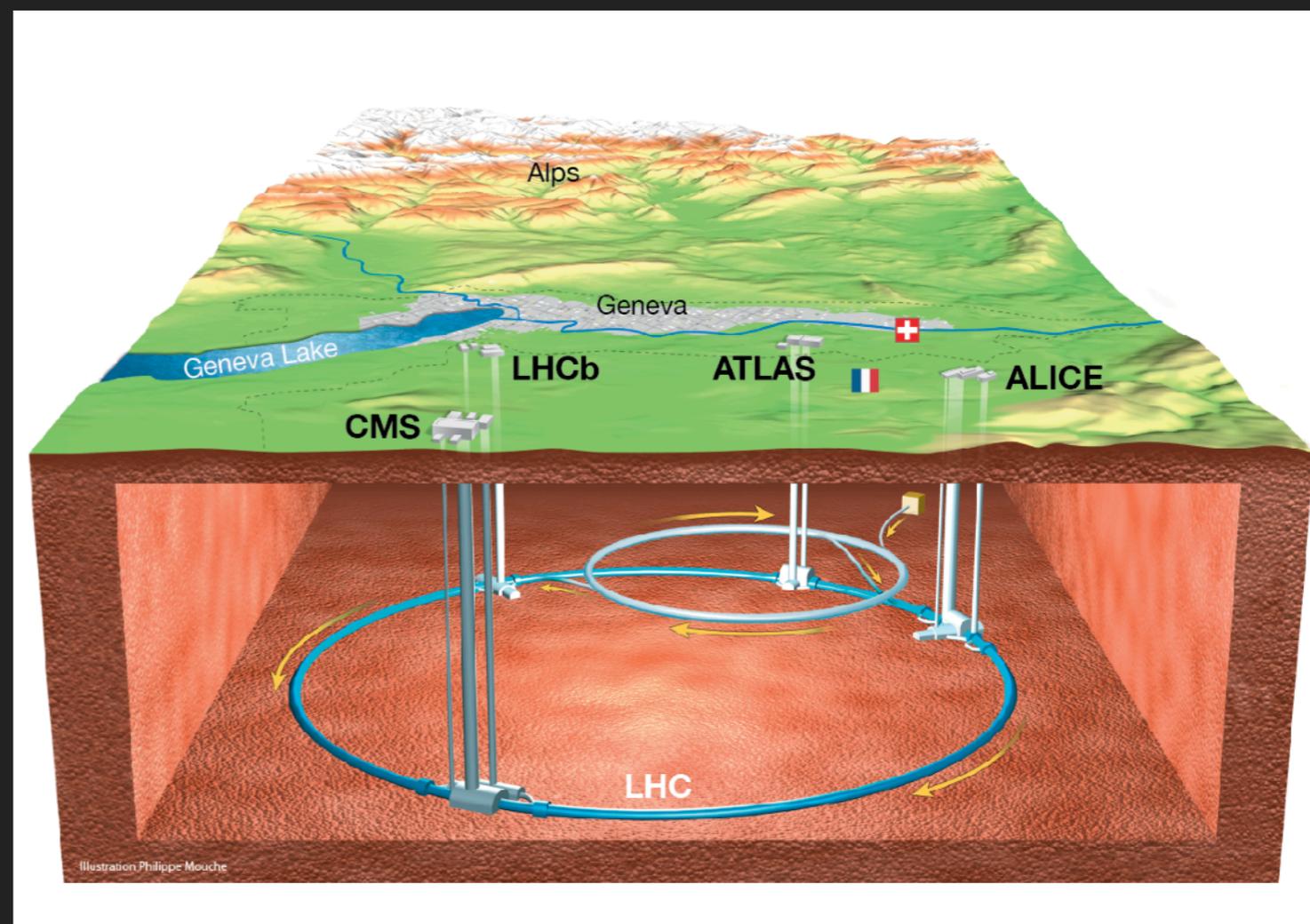
LARGE HADRON COLLIDER & CERN

- ▶ Centro Europeo de investigación Nuclear
- ▶ El laboratorio más grande del mundo

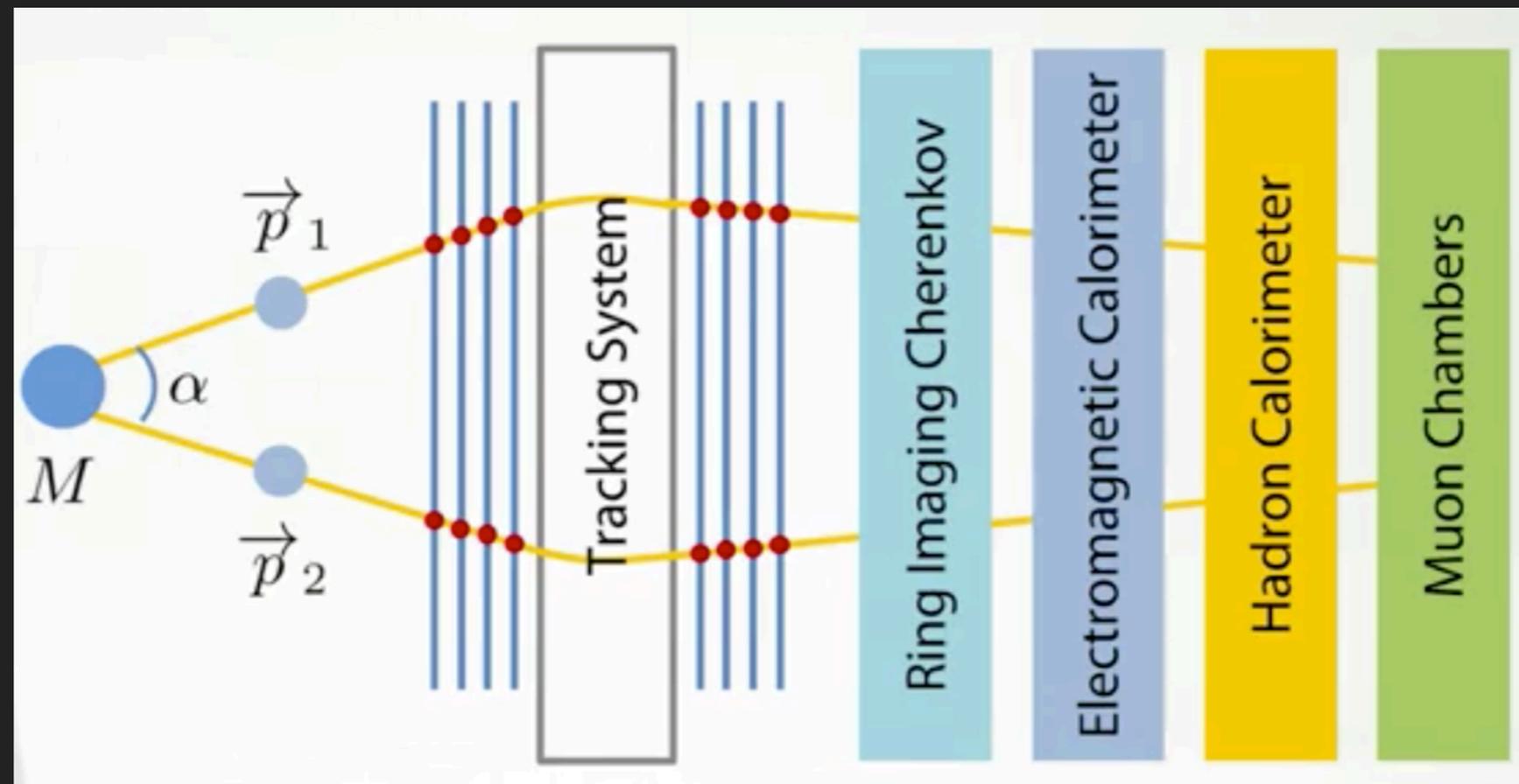
<https://home.cern/science/computing/processing-what-record>

LHC & GRANDES CANTIDADES DE DATOS

- ▶ Túnel 27 km largo, 50-175 m de profundidad
- ▶ Velocidad protones: $0.999999999999 c$
- ▶ 600M eventos/s
- ▶ 100,000 eventos/s enviados para reconstrucción digital
- ▶ 100 a 200 eventos/s procesados
- ▶ 25 GB /s = 90 TB/ hora producido
- ▶ 10 – 100 Petabytes al año

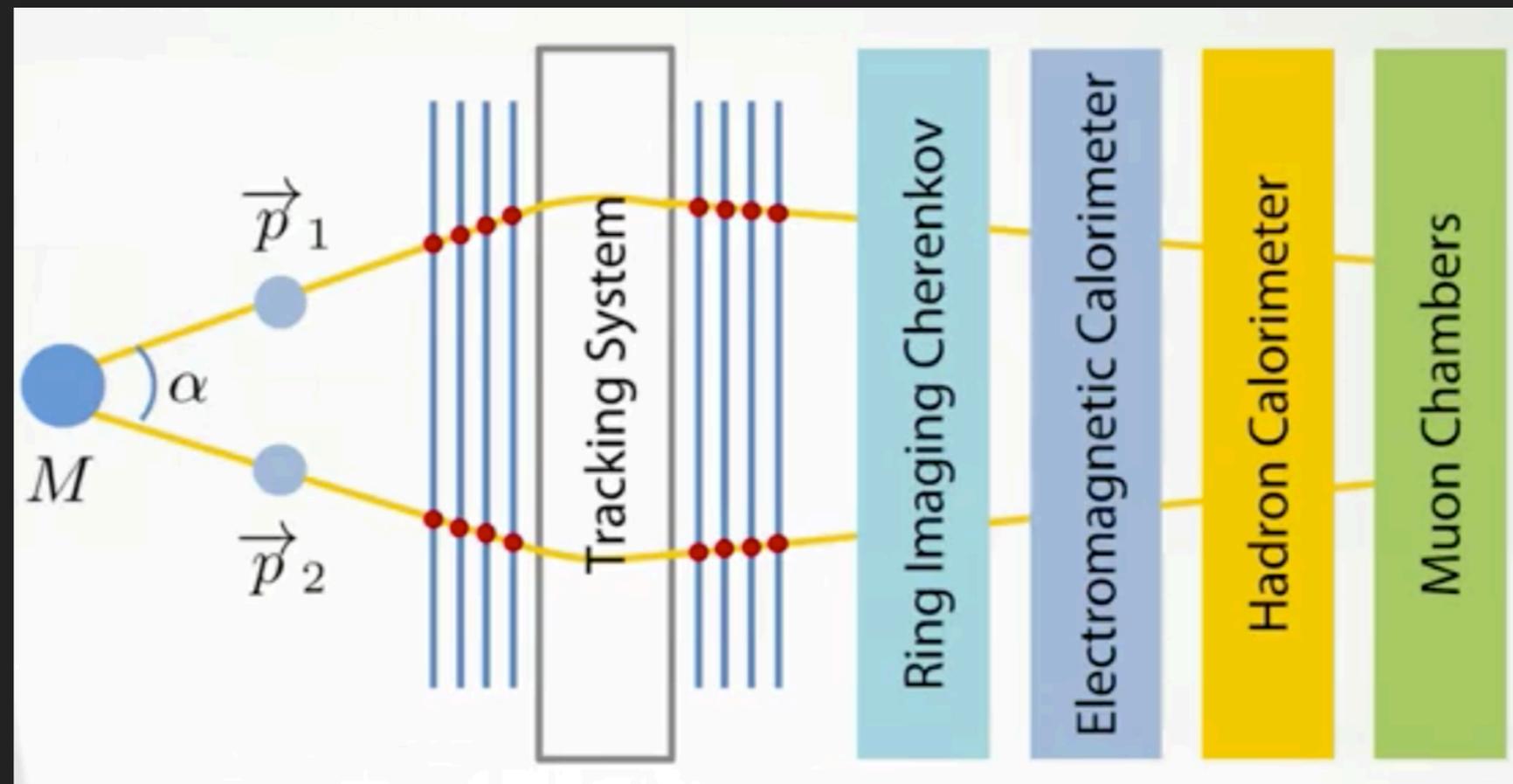


DETECTORES



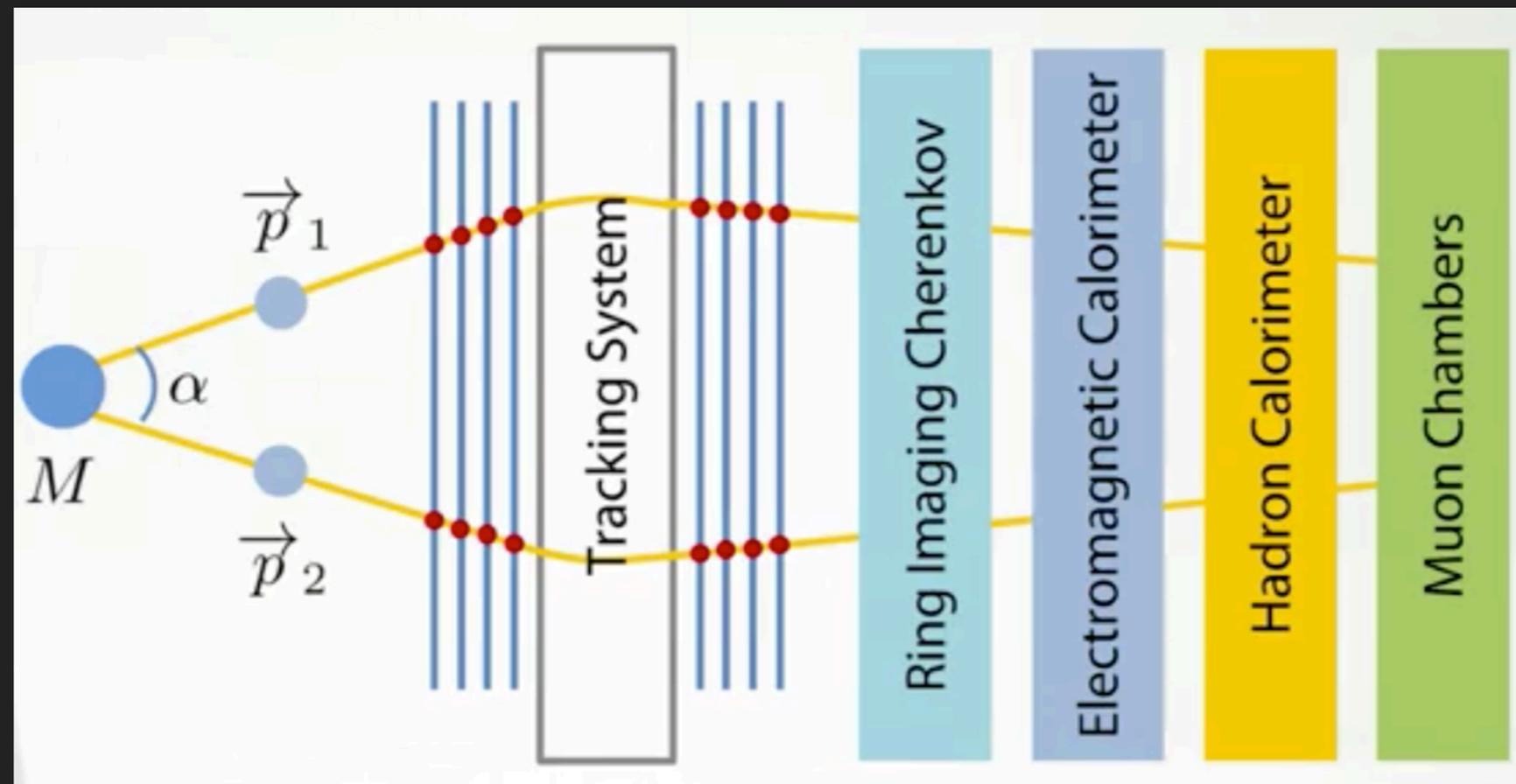
DETECTORES

► Sistema de rastreo



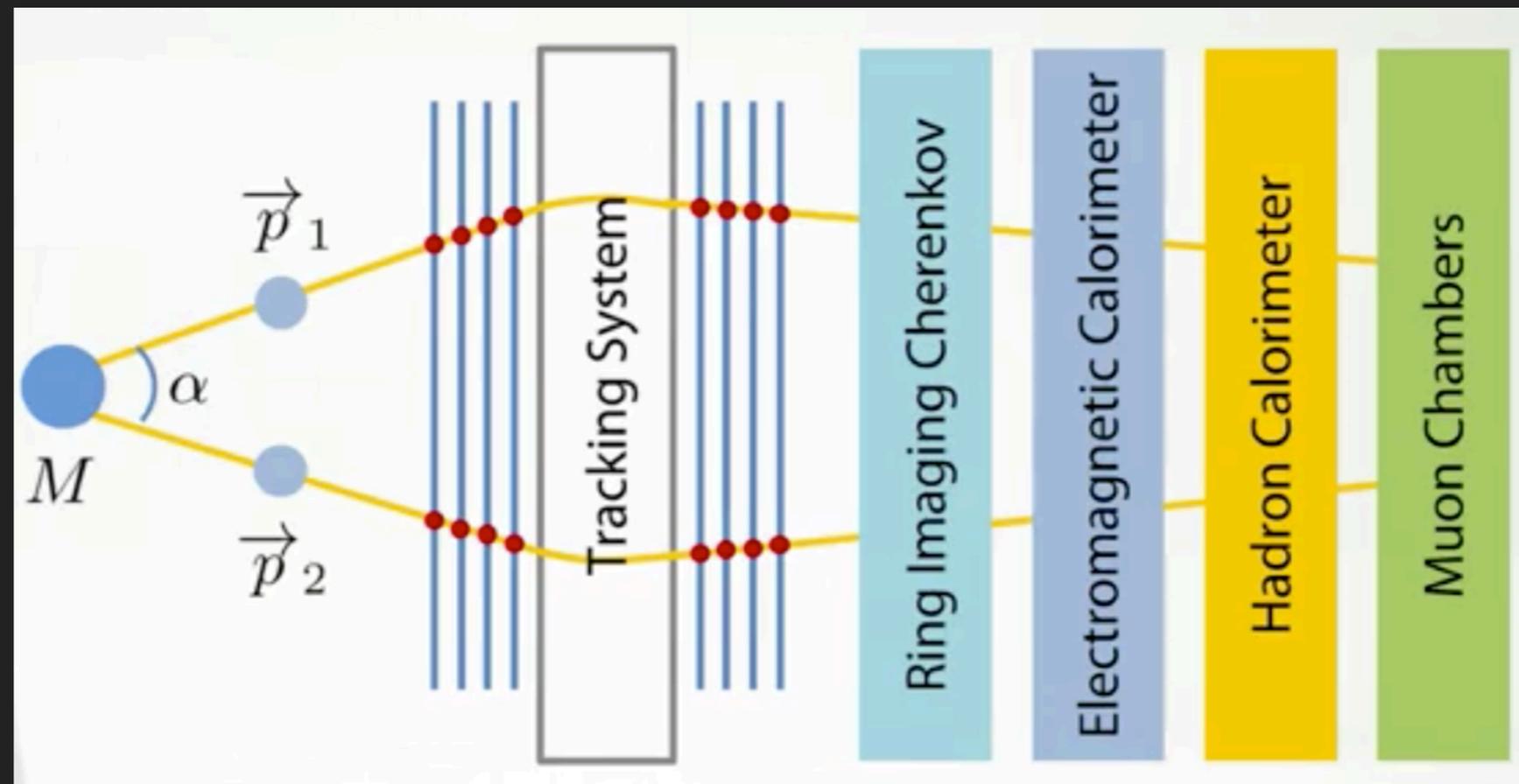
DETECTORES

- ▶ Sistema de rastreo
- ▶ Calorimetro EM



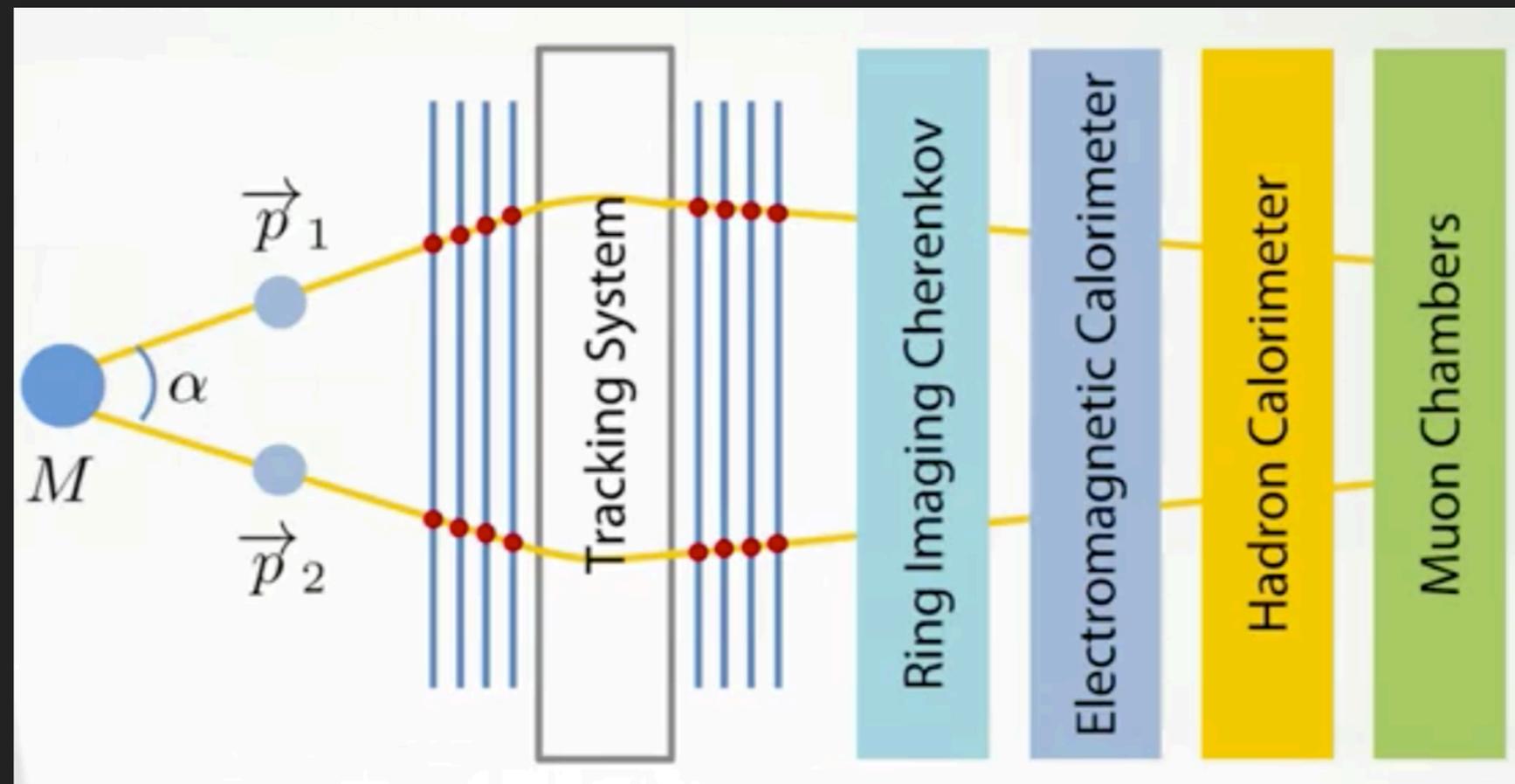
DETECTORES

- ▶ Sistema de rastreo
- ▶ Calorimetro EM
- ▶ Calorimetro H



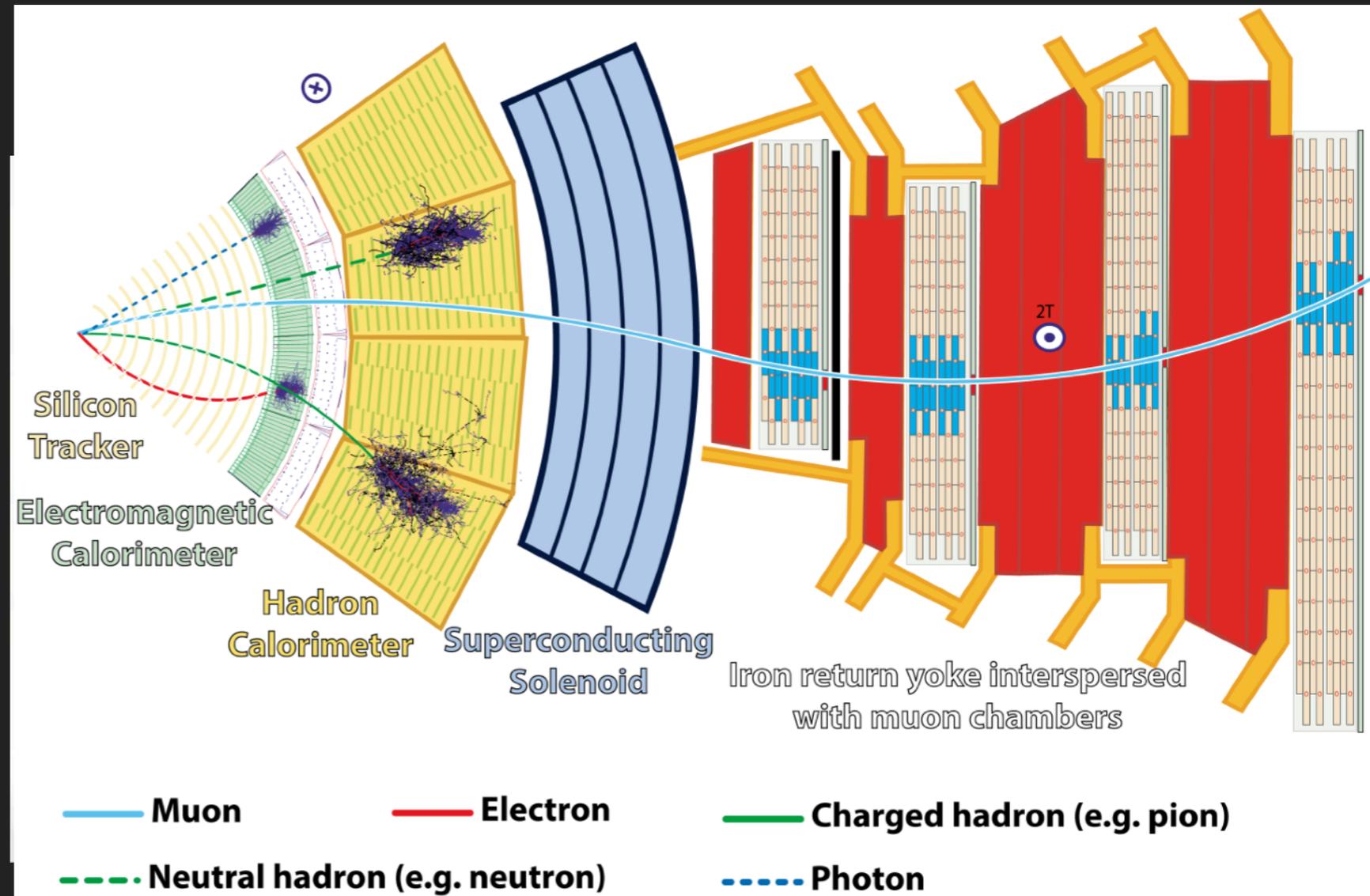
DETECTORES

- ▶ Sistema de rastreo
- ▶ Calorimetro EM
- ▶ Calorimetro H
- ▶ Detectores de Muones



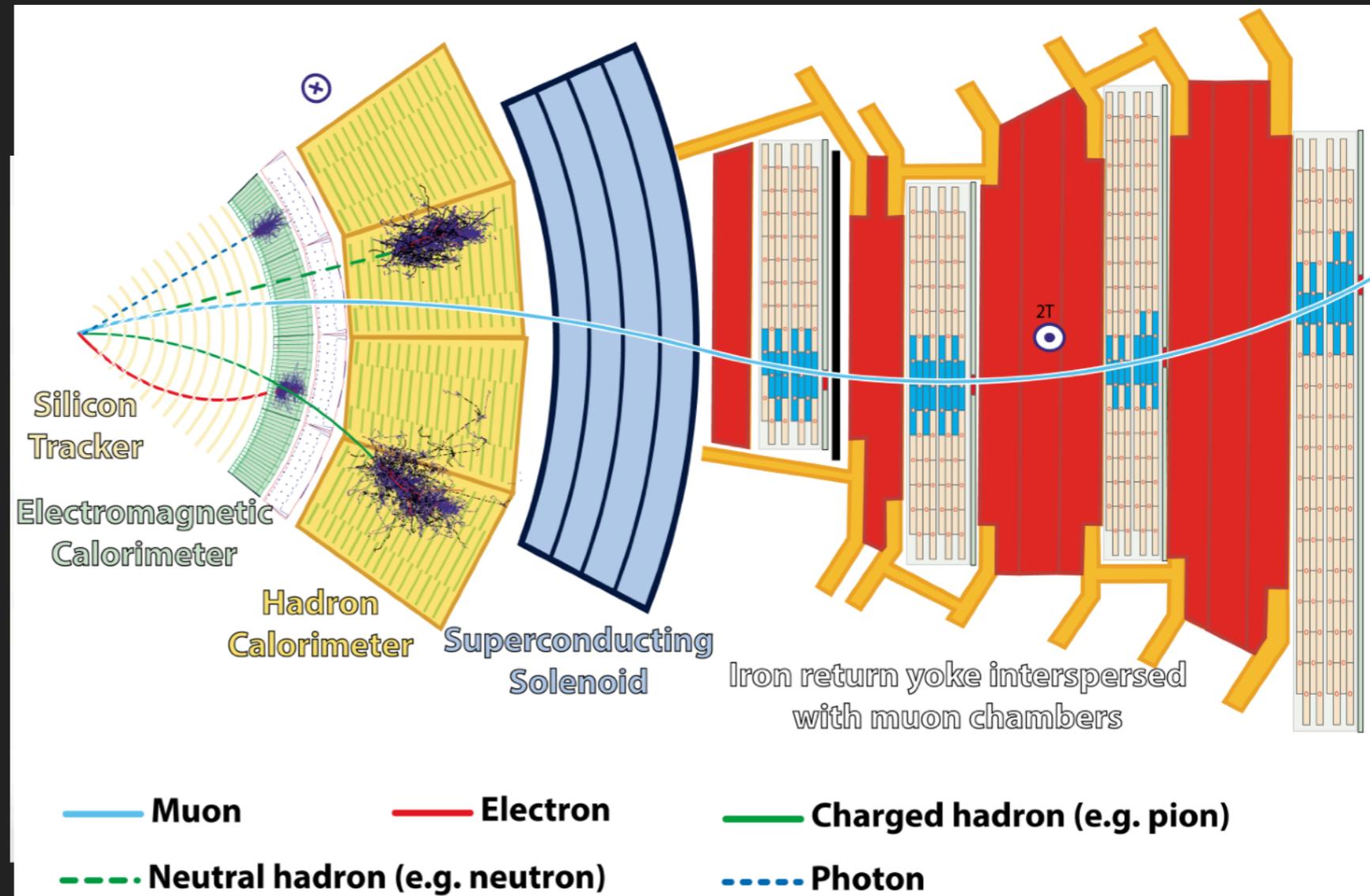
DETECTORES

- ▶ Sistema de rastreo
- ▶ Calorimetro EM
- ▶ Calorimetro H
- ▶ Detectores de Muones



DETECTORES

- ▶ Sistema de rastreo
- ▶ Calorimetro EM
- ▶ Calorimetro H
- ▶ Detectores de Muones

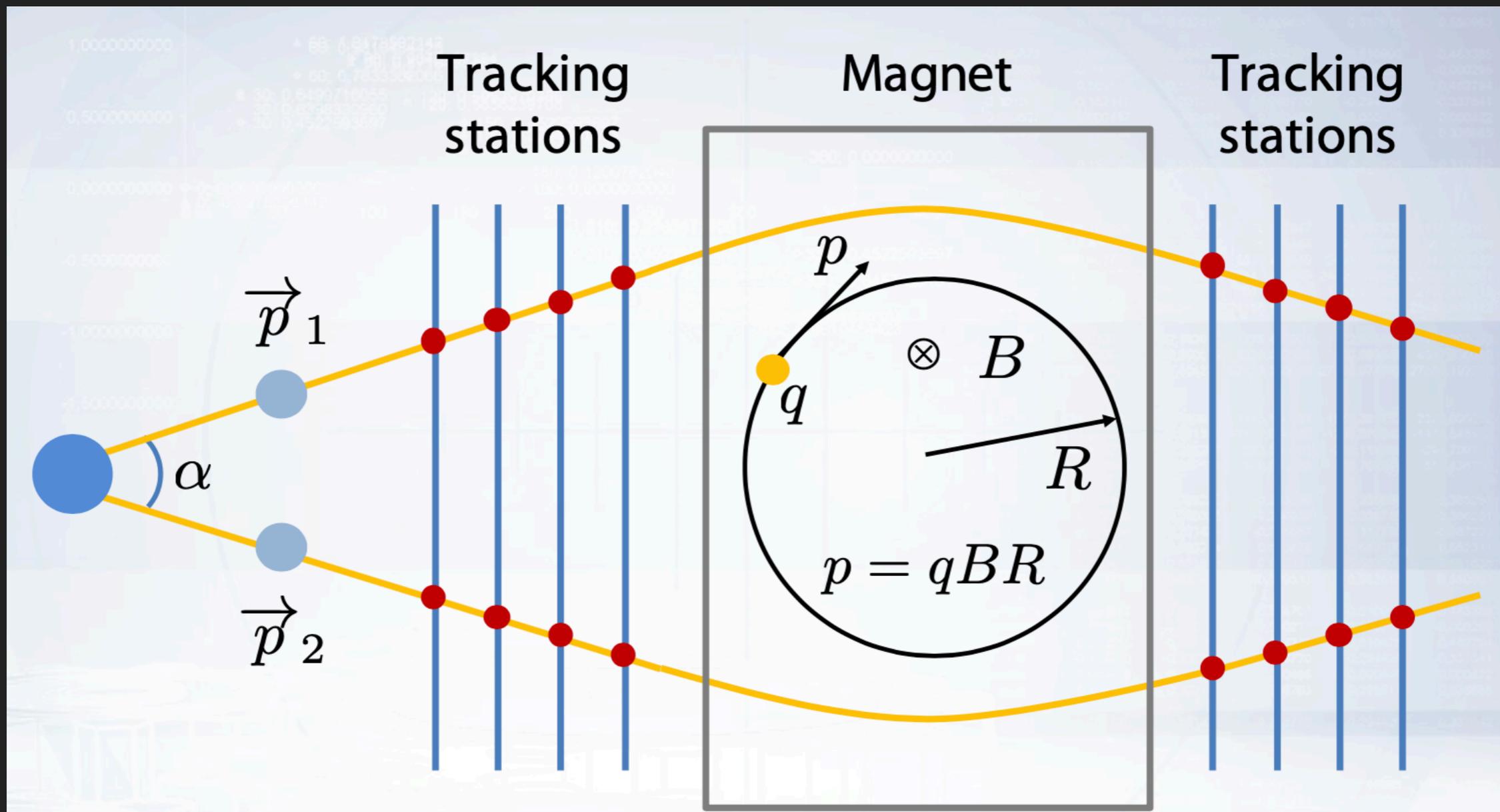


<https://www.coursera.org/learn/hadron-collider-machine-learning/lecture/wZZ75/tracking-system>

<https://cds.cern.ch/record/2120661>

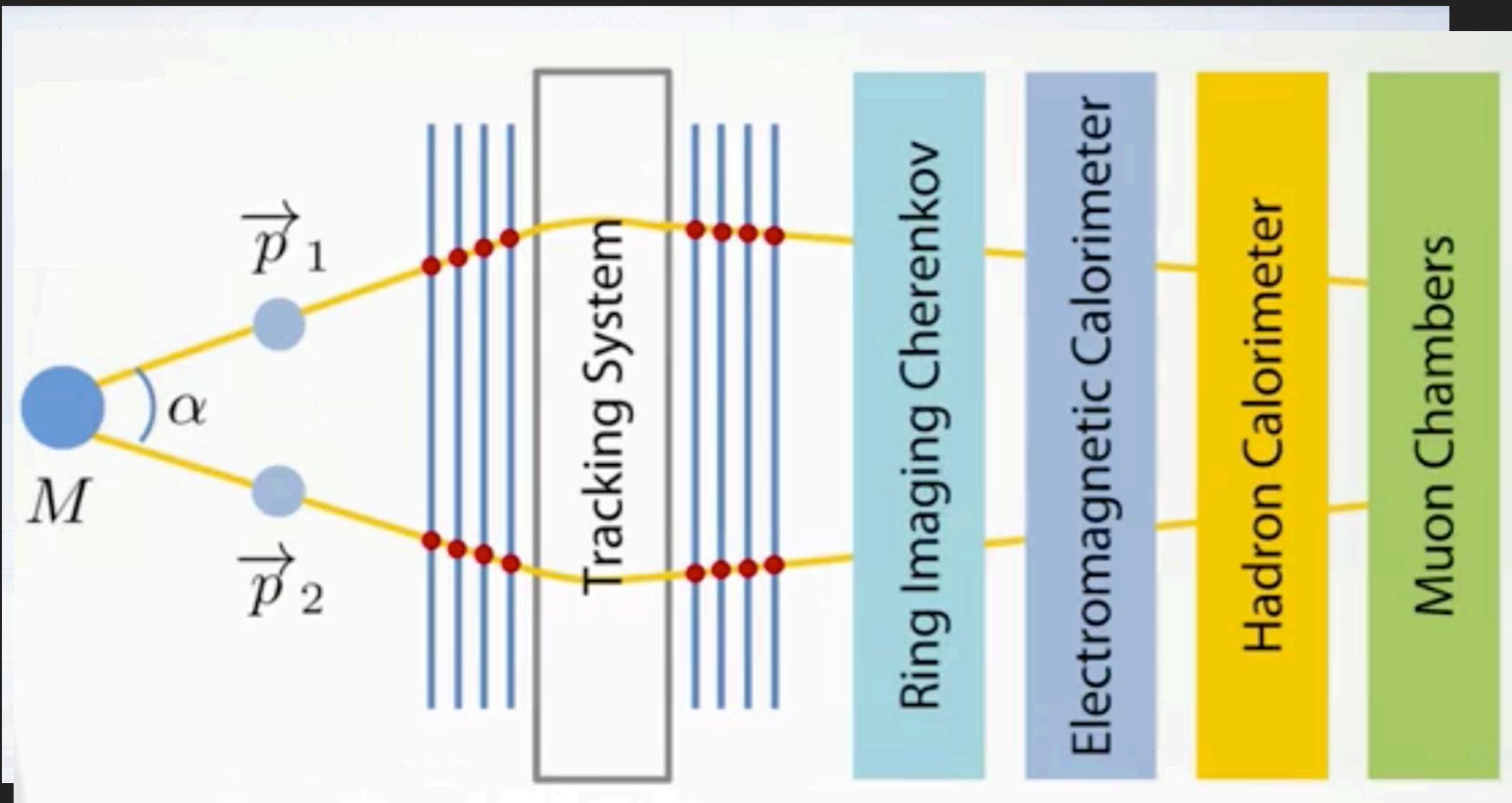
SISTEMA DE RASTREO

- ▶ Sensores - silicon trackers (Píxeles de silicio)



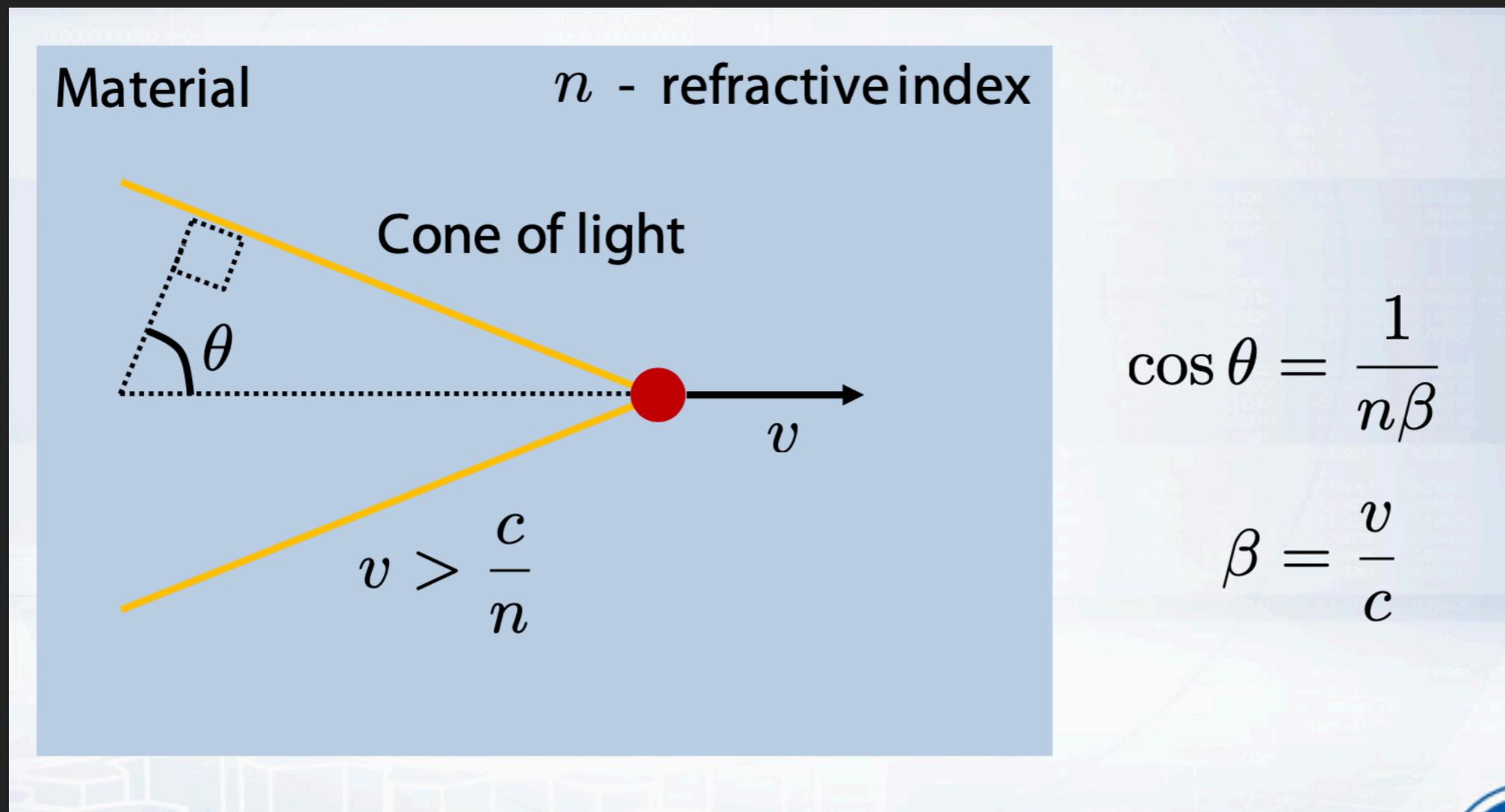
RING IMAGING CHERENKOV (RICH) DETECTOR

- ▶ Radiación de Cherenkov
- ▶ Identificación de partículas



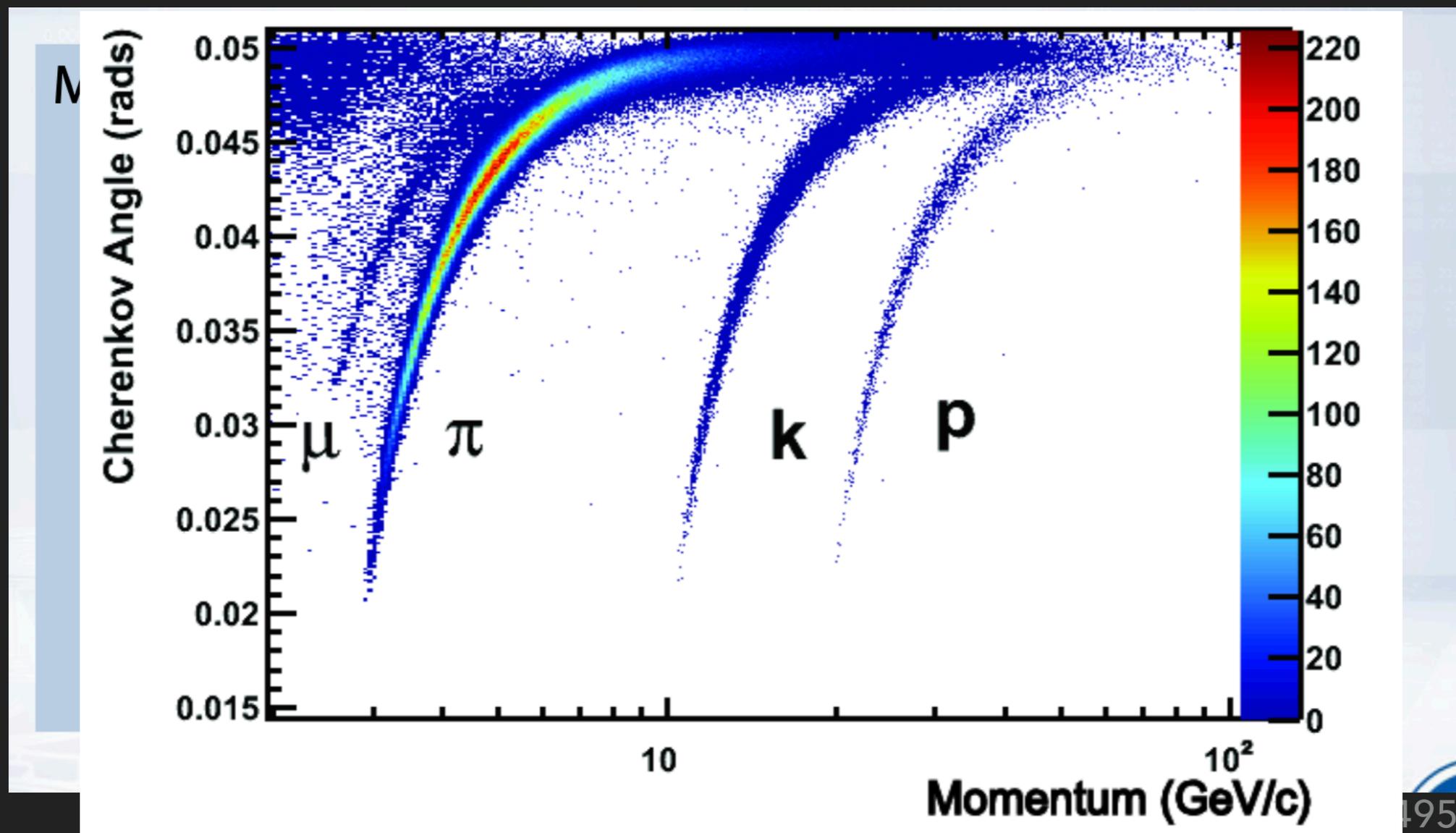
RING IMAGING CHERENKOV (RICH) DETECTOR

- ▶ Radiación de Cherenkov
- ▶ Identificación de partículas



RING IMAGING CHERENKOV (RICH) DETECTOR

- ▶ Radiación de Cherenkov
- ▶ Identificación de partículas



CALORIMETRO ELETTROMAGNETICO

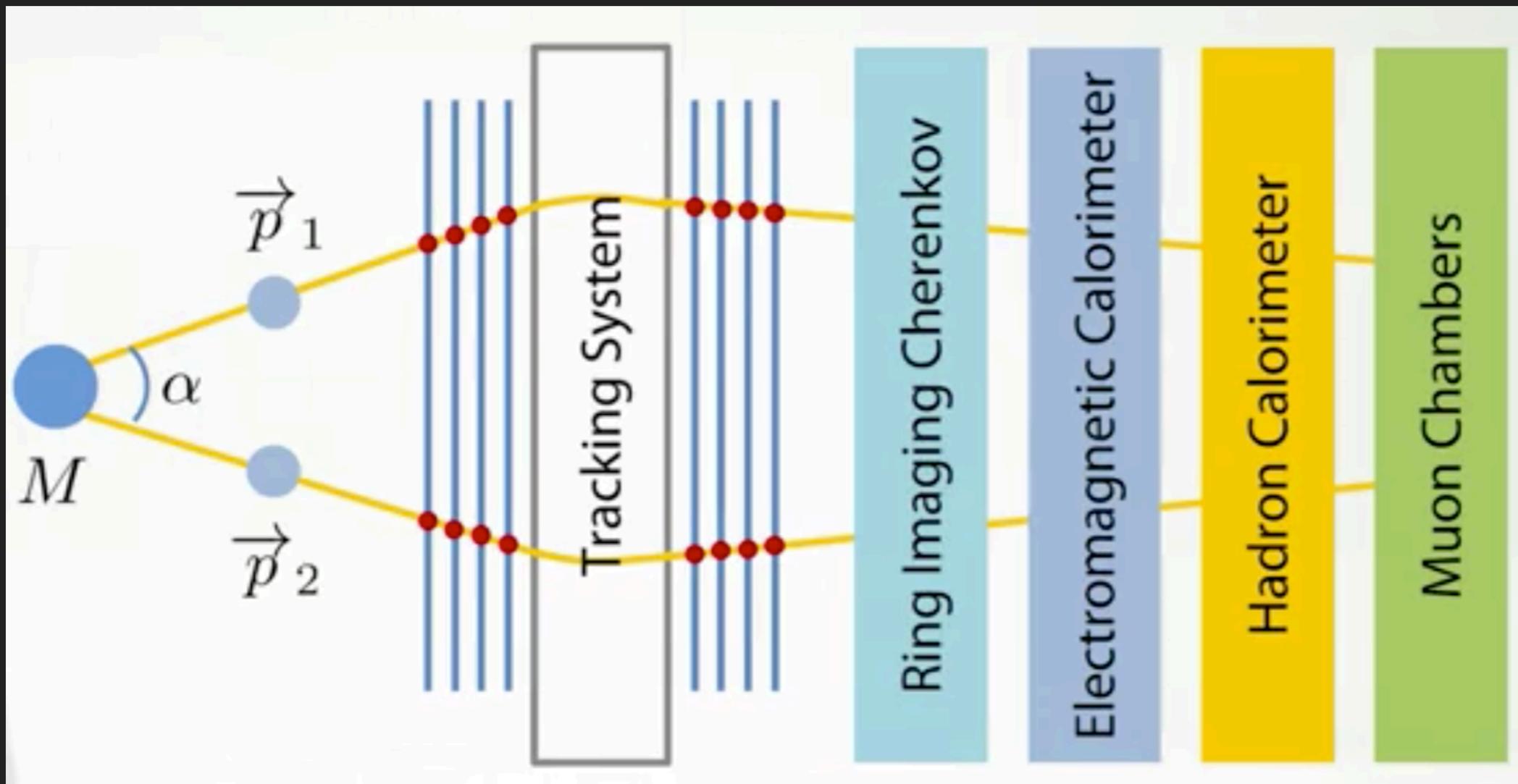
- ▶ Mide la energía de electrones y fotones

<http://cds.cern.ch/record/835712>

CALORIMETRO ELETTRONMAGNETICO

- Mide la energía de electrones y fotones

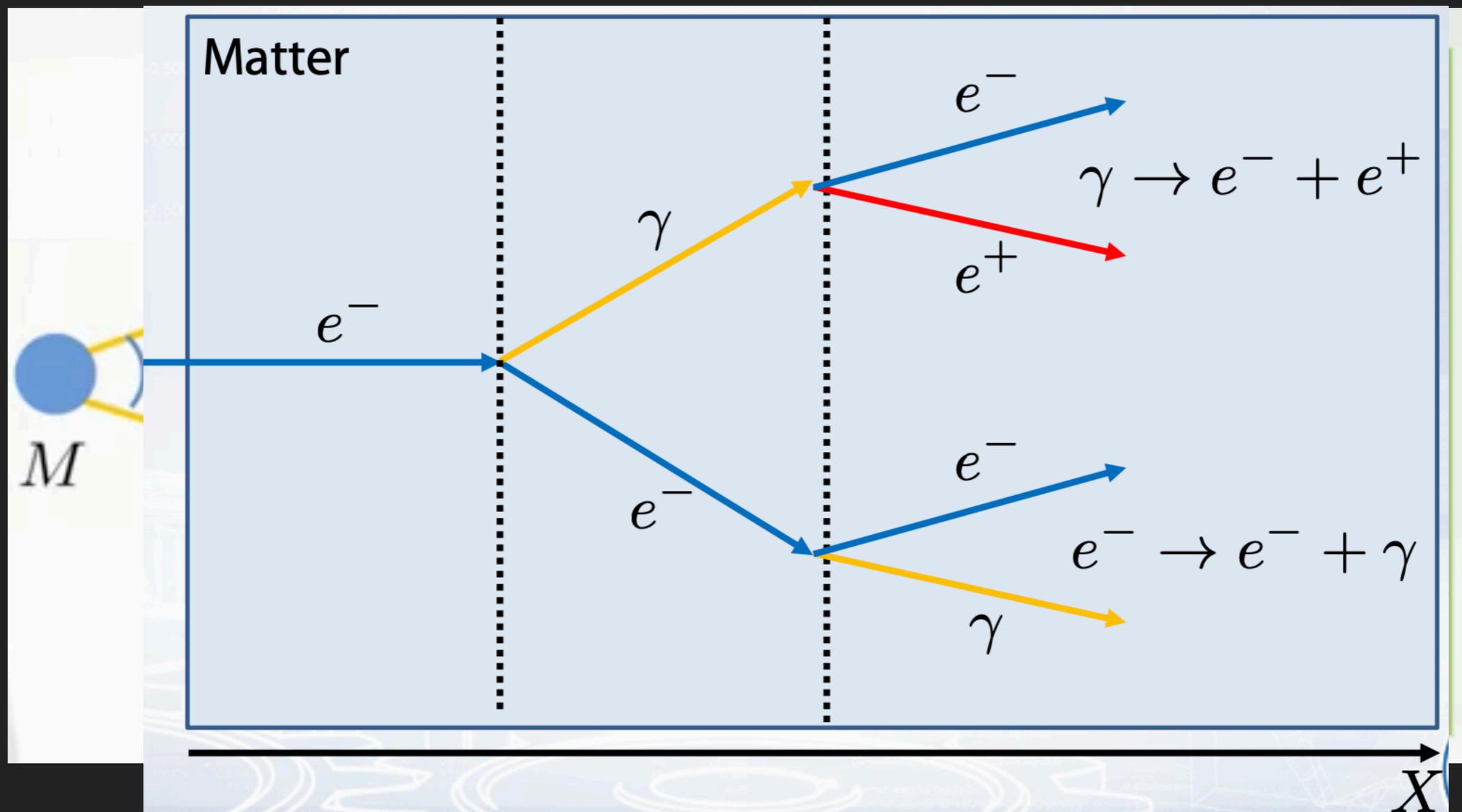
<http://cds.cern.ch/record/835712>



CALORIMETRO ELETTRIMAGNETICO

- Mide la energía de electrones y fotones

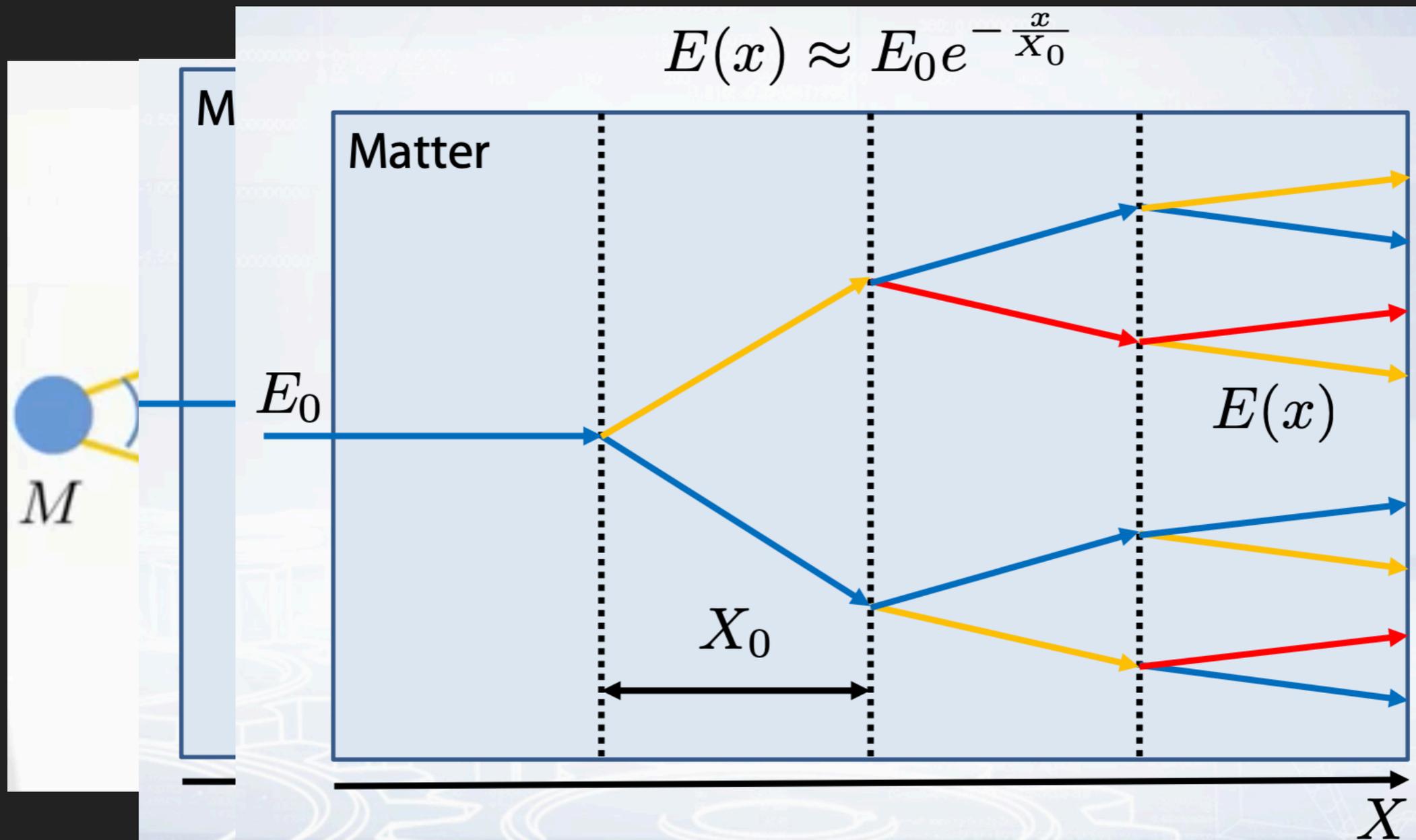
<http://cds.cern.ch/record/835712>



CALORIMETRO ELETTRONMAGNETICO

- Mide la energía de electrones y fotones

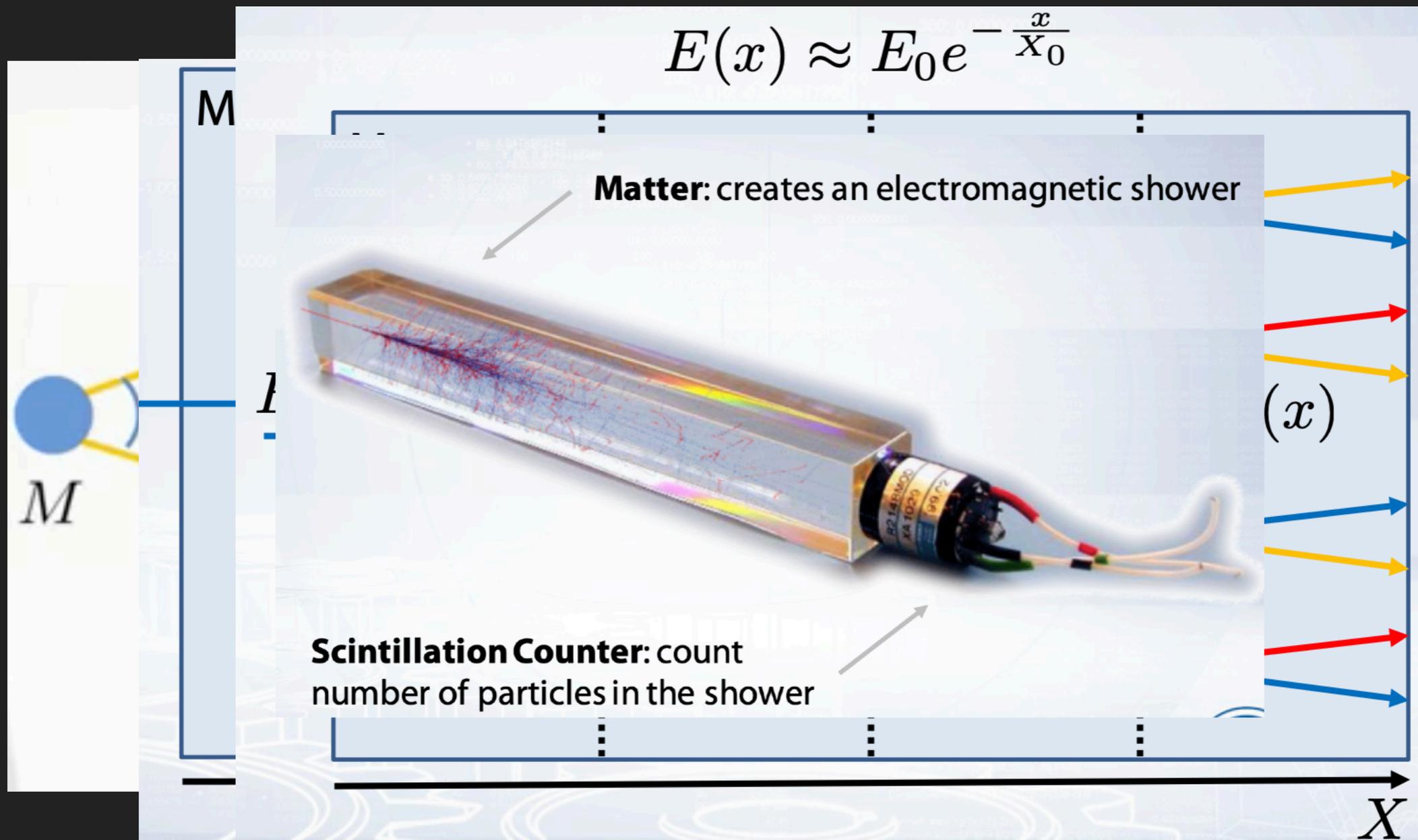
<http://cds.cern.ch/record/835712>



CALORIMETRO ELETTRIMAGNETICO

- Mide la energía de electrones y fotones

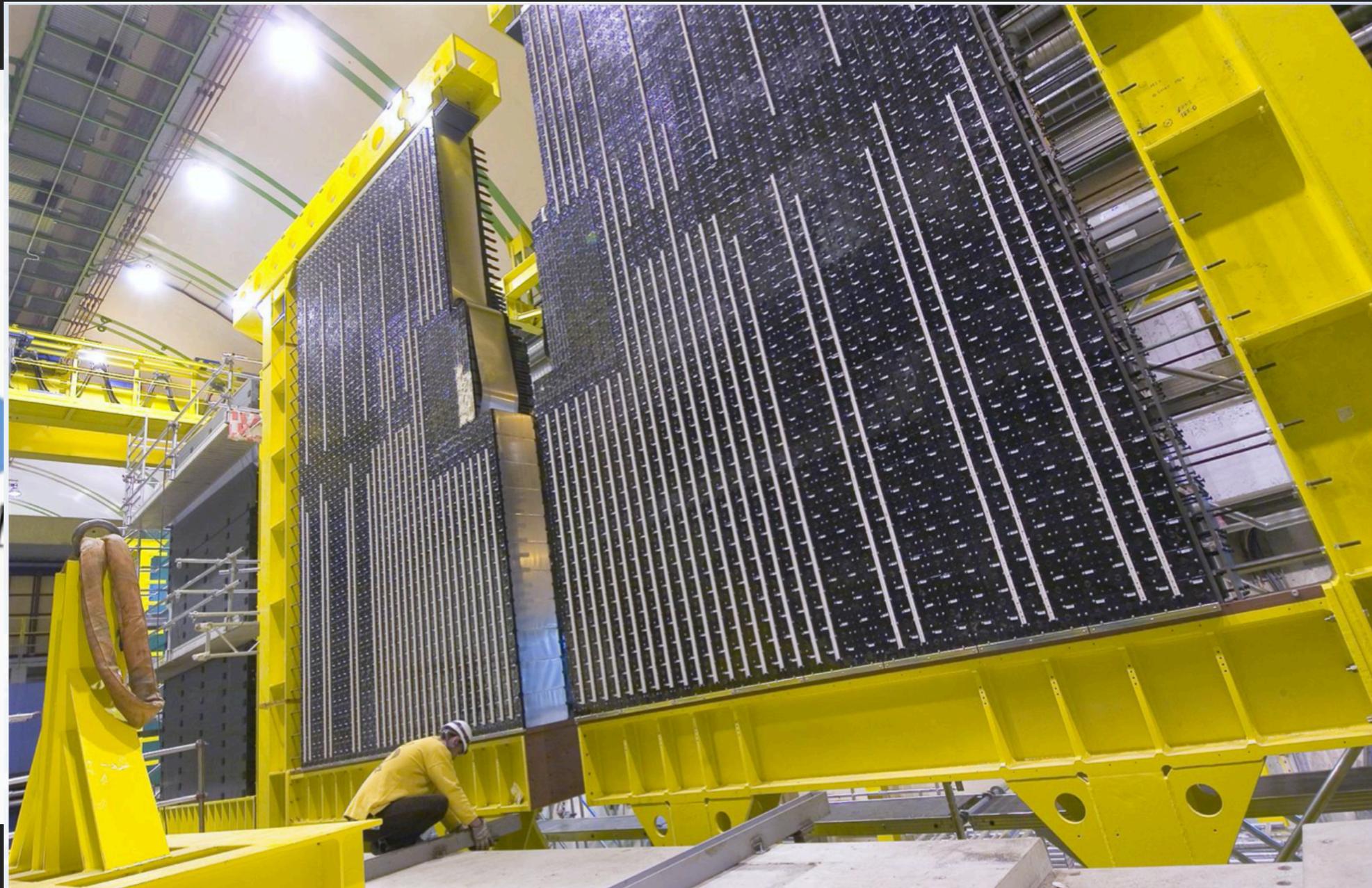
<http://cds.cern.ch/record/835712>



CALORIMETRO ELETTRONMAGNETICO

- Mide la energía de electrones y fotones

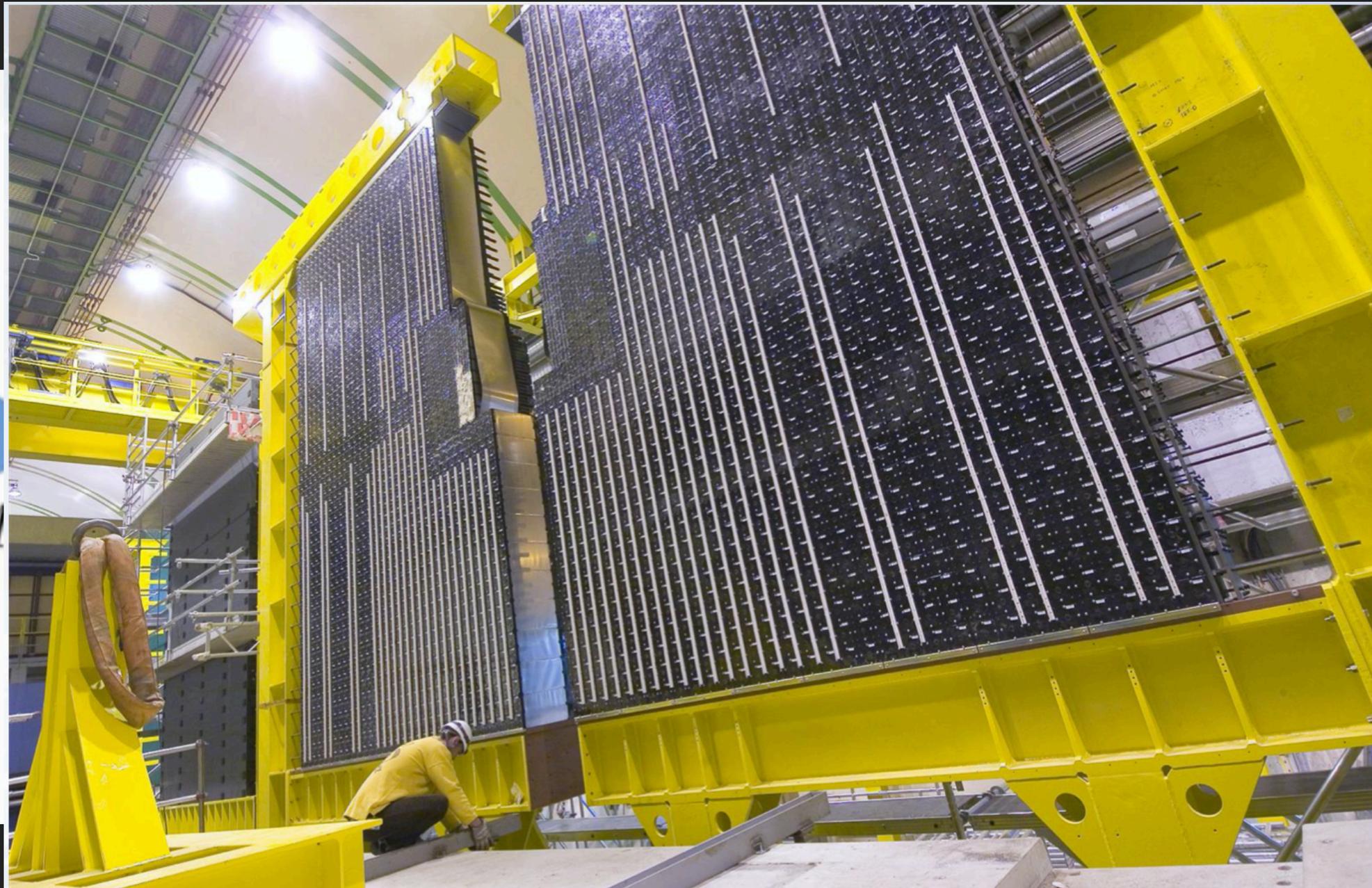
<http://cds.cern.ch/record/835712>



CALORIMETRO ELETTRONMAGNETICO

- Mide la energía de electrones y fotones

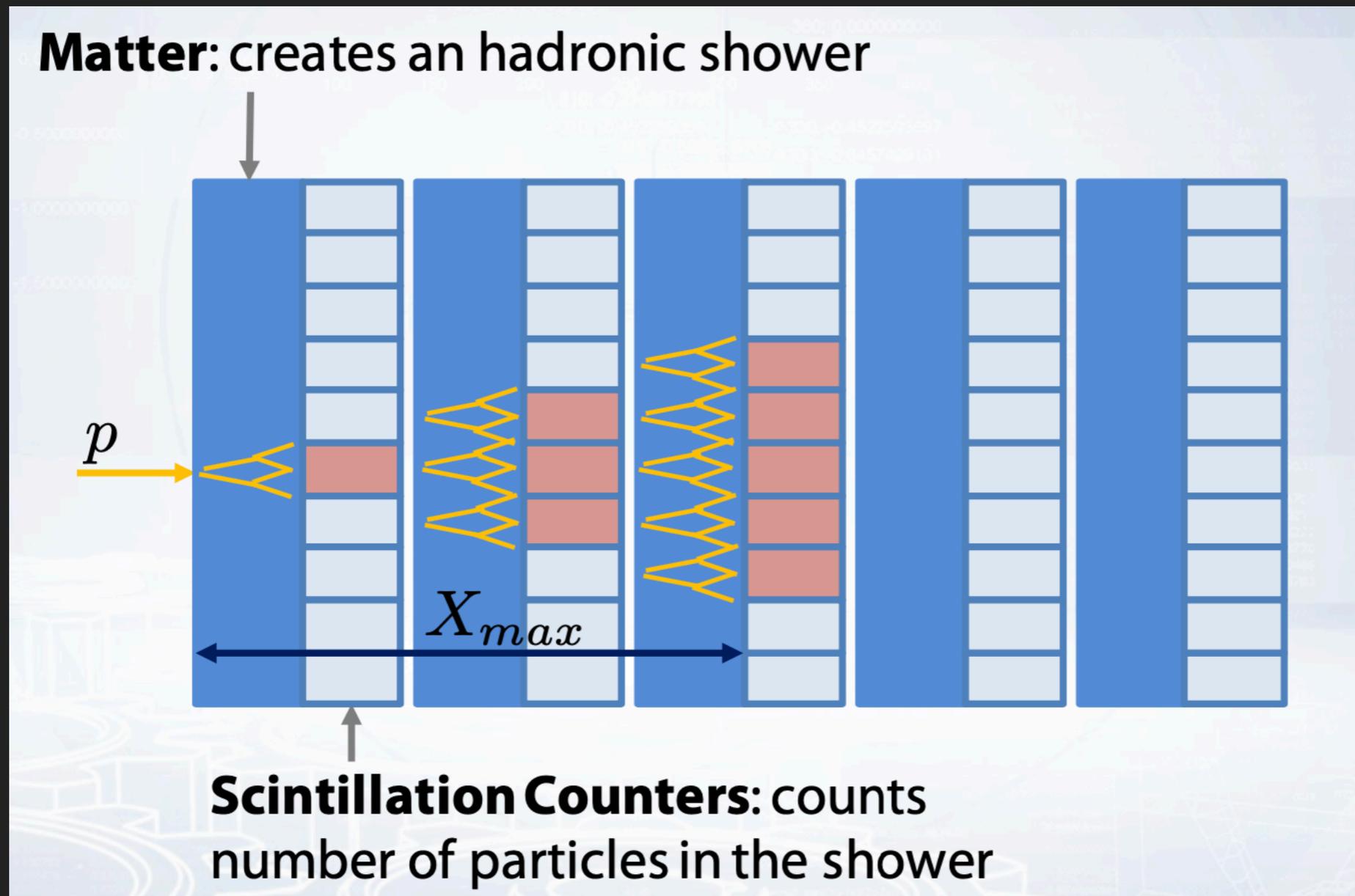
<http://cds.cern.ch/record/835712>



LHCb

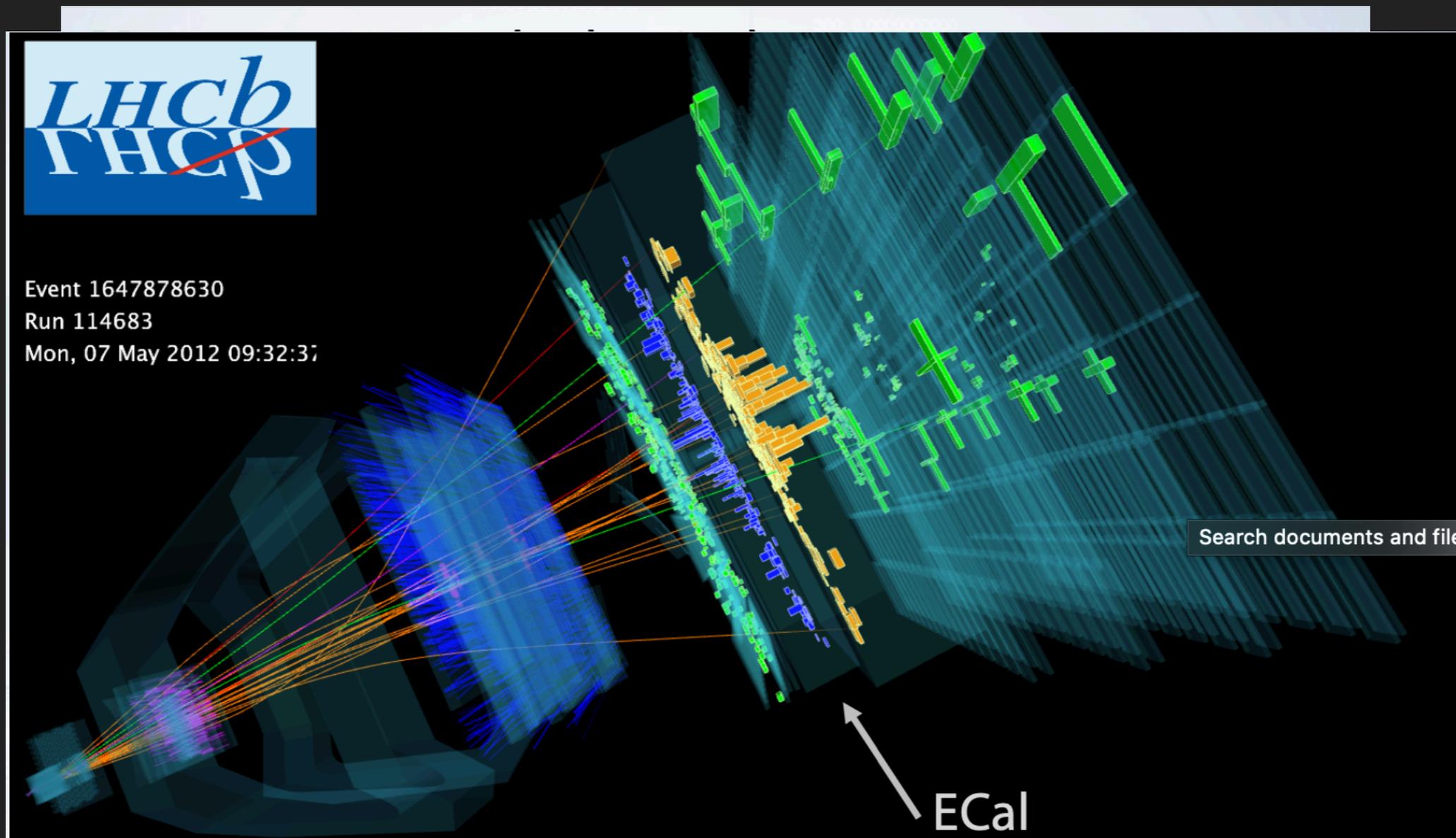
CALORIMETRO HADRONICO

- ▶ Mide la energía de Hadrones



CALORIMETRO HADRONICO

- ▶ Mide la energía de Hadrones



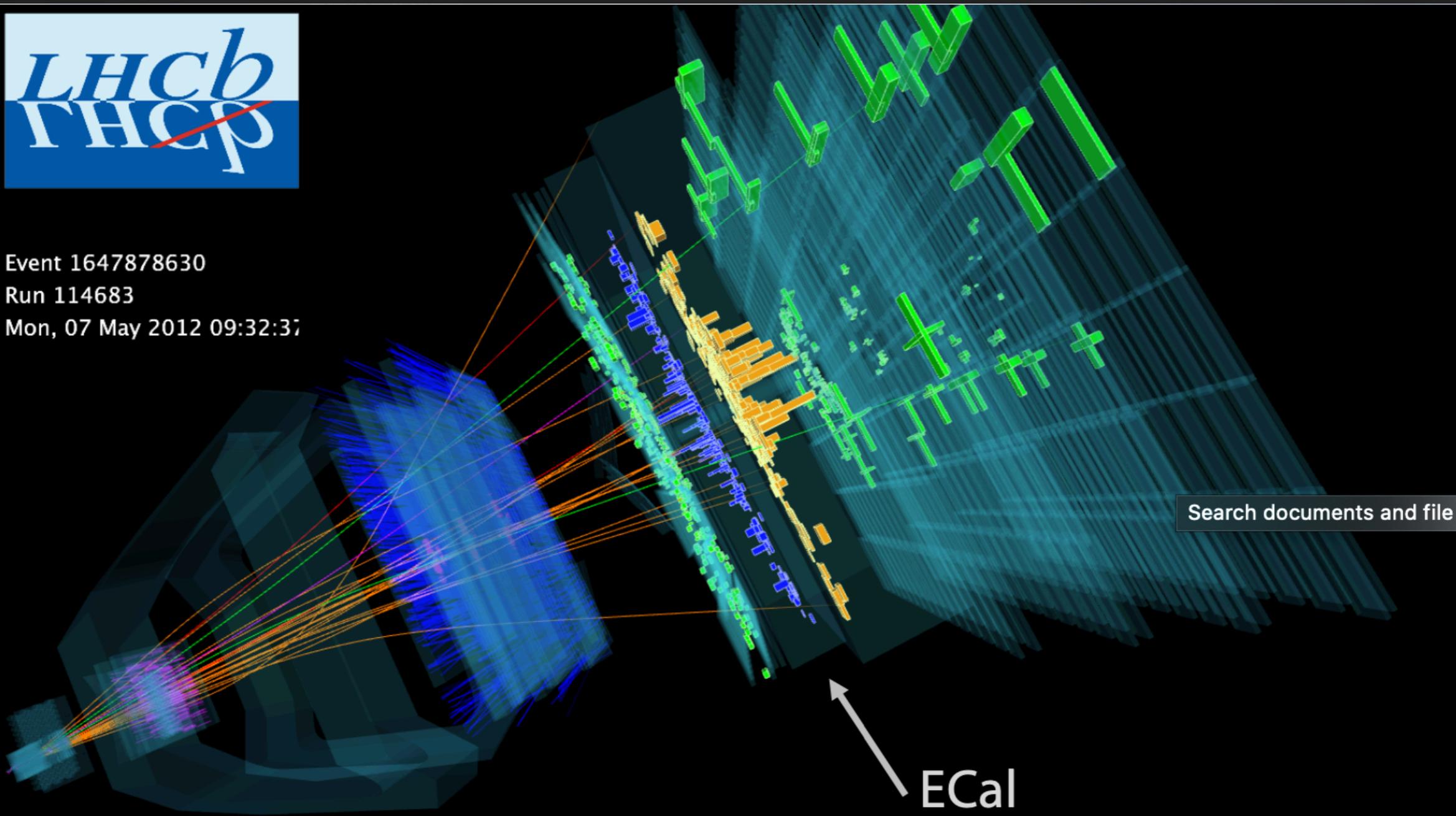
DETECTORES

DETECTORES DE MUONES

DETECTORES DE MUONES



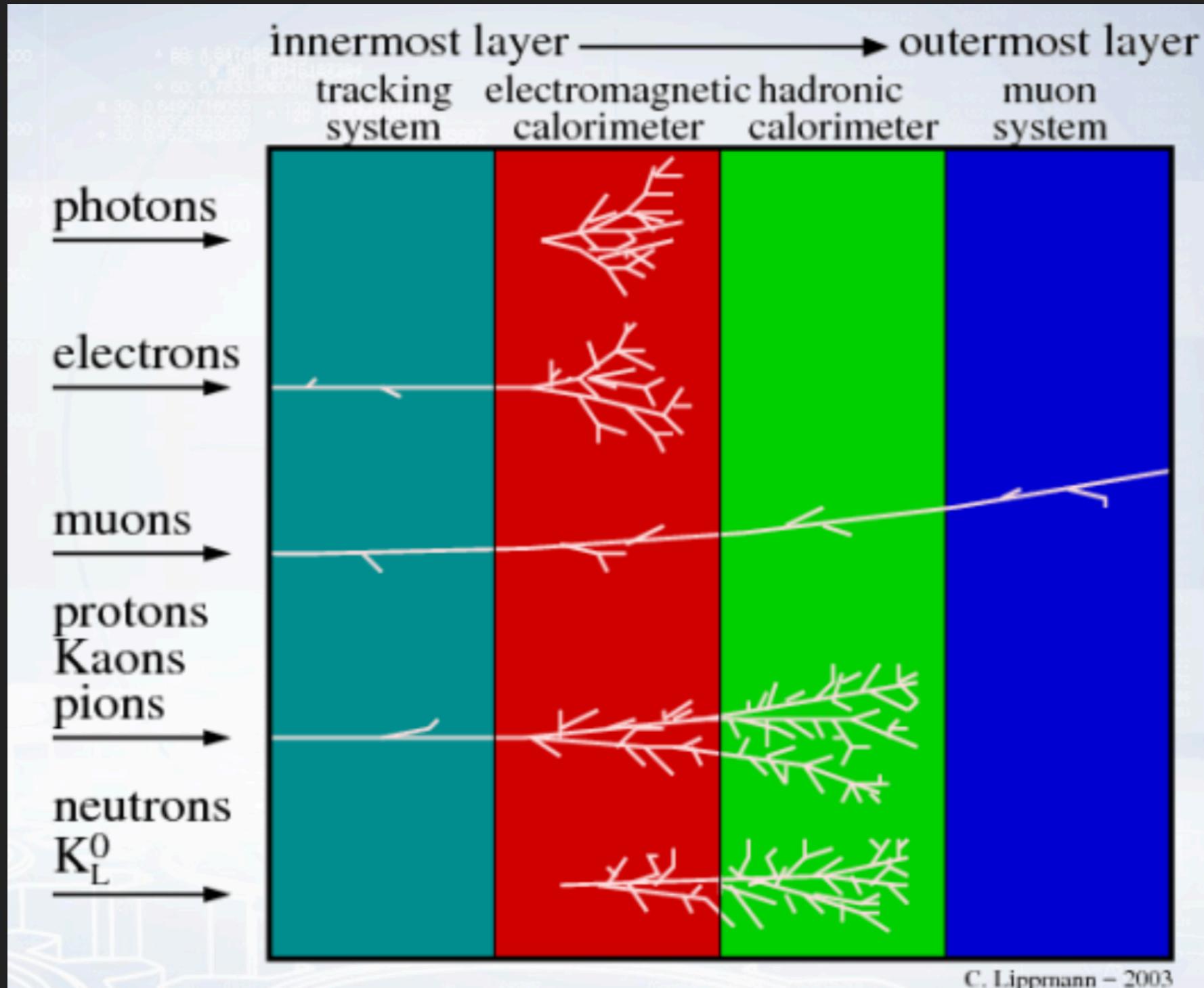
Event 1647878630
Run 114683
Mon, 07 May 2012 09:32:37



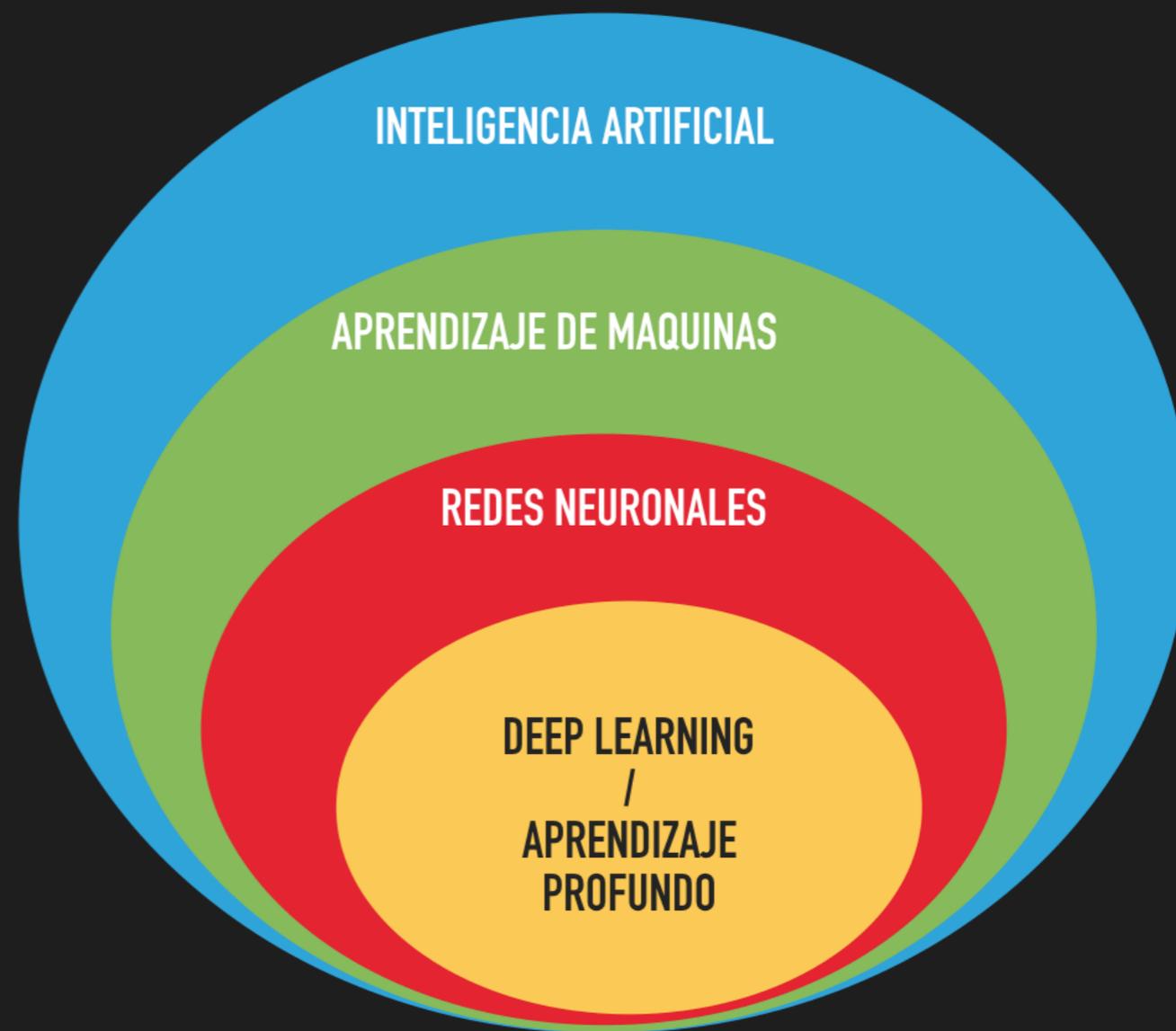
ECal

Search documents and file

DETECTORES

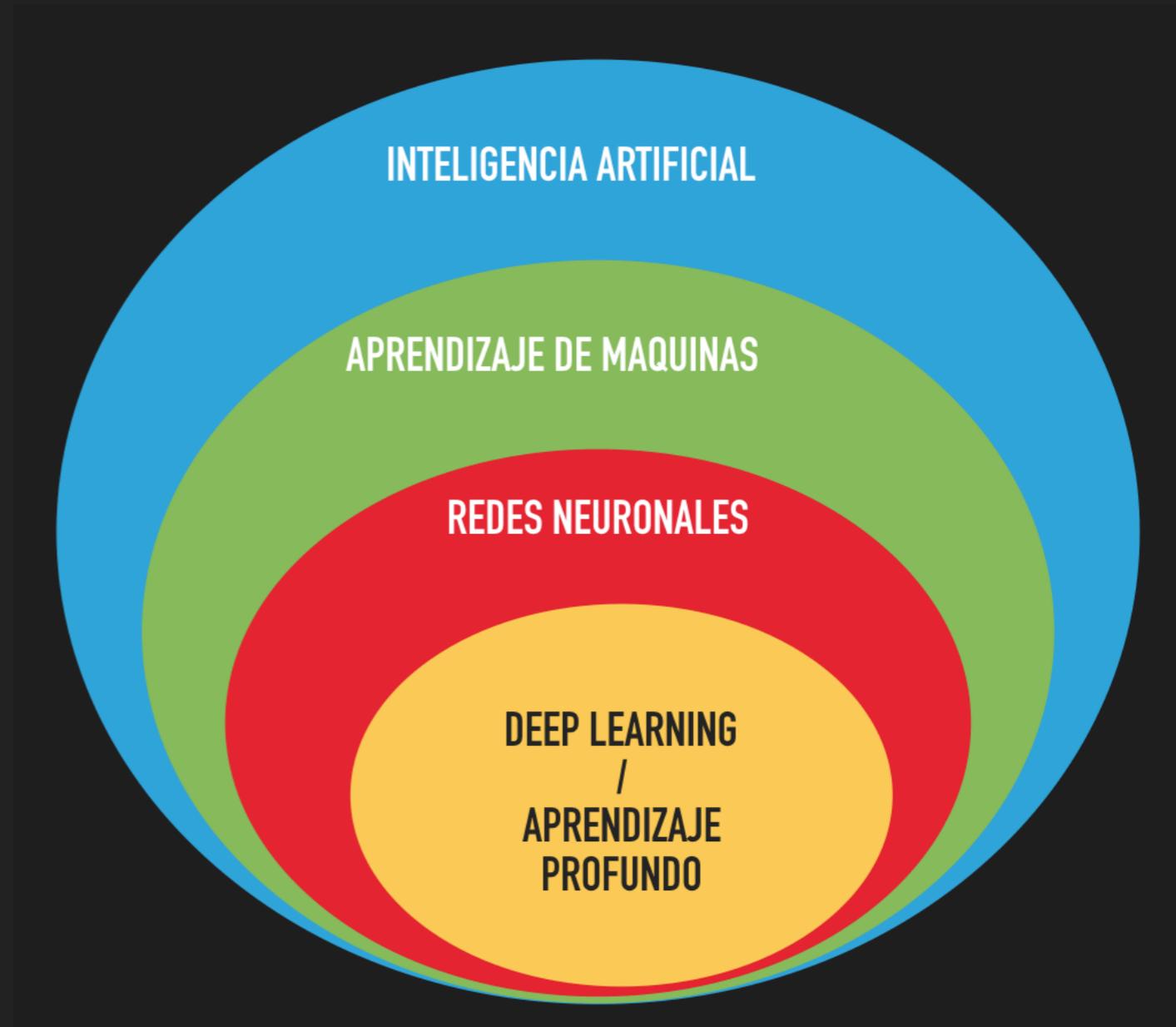


MACHINE LEARNING



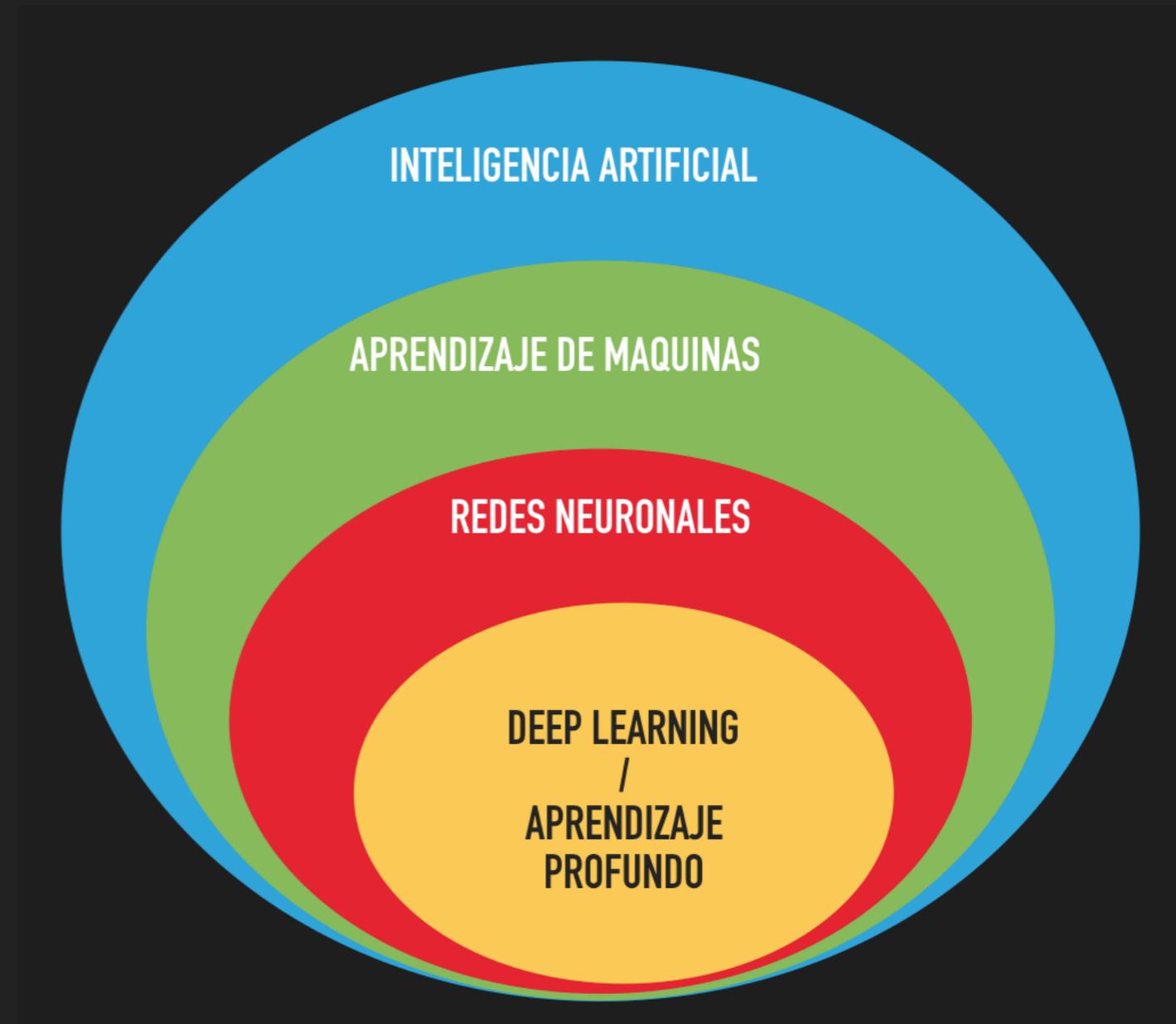
MACHINE LEARNING

▶ Aprendizaje Supervisado



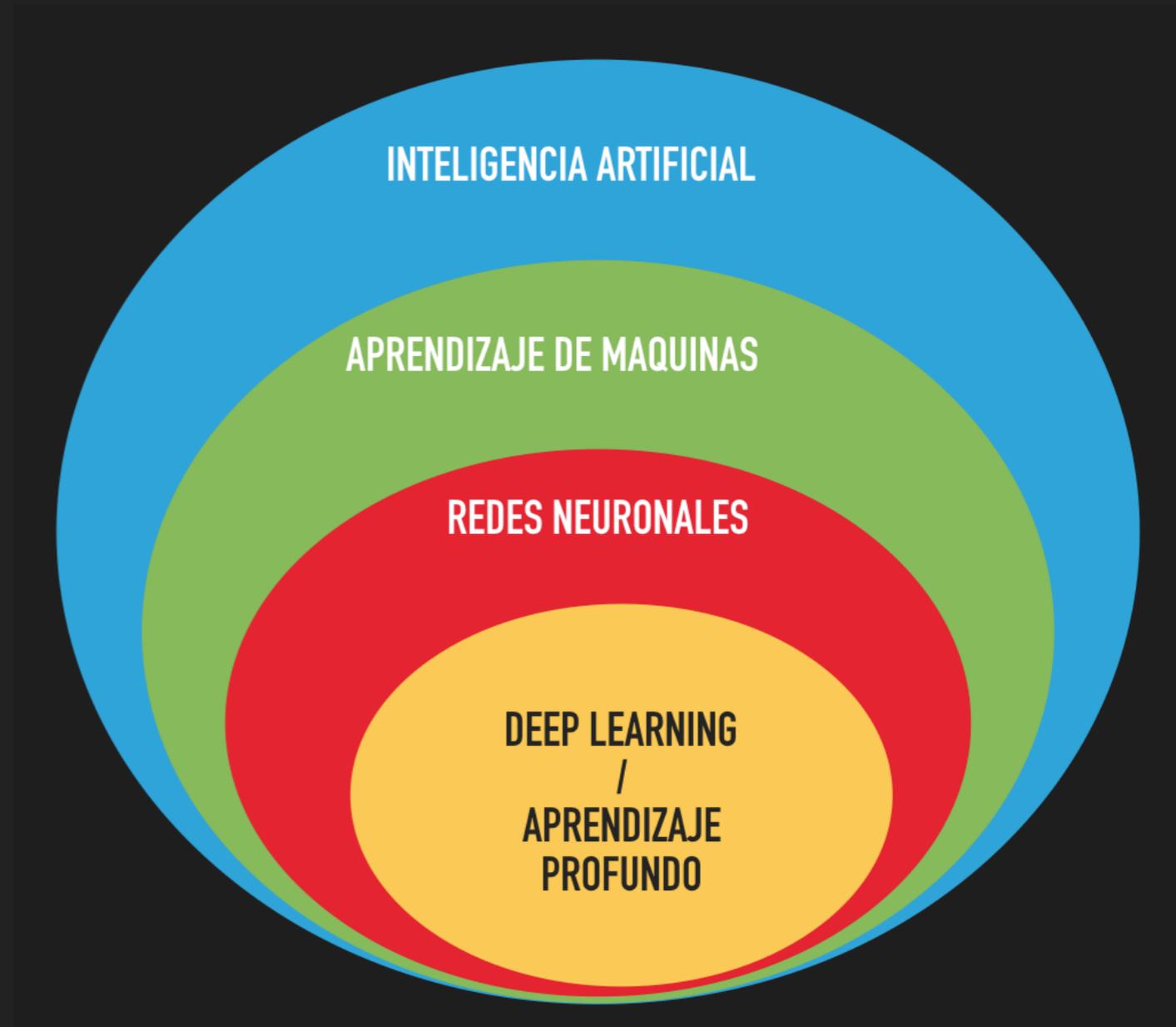
MACHINE LEARNING

- ▶ Aprendizaje Supervisado
- ▶ No - Supervisado



MACHINE LEARNING

- ▶ Aprendizaje Supervisado
- ▶ No - Supervisado
- ▶ Aprendizaje por Refuerzo



APRENDIZAJE SUPERVISADO

APRENDIZAJE SUPERVISADO

- ▶ n muestras de datos

$$\mathbf{x}_i \in \mathbb{R}^p; i = 1, \dots, n$$

APRENDIZAJE SUPERVISADO

- ▶ n muestras de datos

$$\mathbf{x}_i \in \mathbb{R}^p; i = 1, \dots, n$$

- ▶ n etiquetas $\mathbf{y}_i \in \mathbb{R}^d; i = 1, \dots, n$

APRENDIZAJE SUPERVISADO

- ▶ n muestras de datos

$$\mathbf{x}_i \in \mathbb{R}^p; i = 1, \dots, n$$

- ▶ n etiquetas $\mathbf{y}_i \in \mathbb{R}^d; i = 1, \dots, n$

- ▶ Conjunto de datos (training

$$\text{set}) \mathcal{D} = \{\mathbf{x}, \mathbf{y}\}_{i=1, \dots, n}$$

APRENDIZAJE SUPERVISADO

- ▶ n muestras de datos

$$\mathbf{x}_i \in \mathbb{R}^p; i = 1, \dots, n$$

- ▶ n etiquetas $\mathbf{y}_i \in \mathbb{R}^d; i = 1, \dots, n$

- ▶ Conjunto de datos (training

$$\text{set}) \mathcal{D} = \{\mathbf{x}, \mathbf{y}\}_{i=1, \dots, n}$$

- ▶ aprender $\mathbf{f}_{\mathbf{w}} : \mathbf{x} \longrightarrow \mathbf{y}$

APRENDIZAJE SUPERVISADO

- ▶ n muestras de datos

$$\mathbf{x}_i \in \mathbb{R}^p; i = 1, \dots, n$$

- ▶ n etiquetas $\mathbf{y}_i \in \mathbb{R}^d; i = 1, \dots, n$

- ▶ Conjunto de datos (training

$$\text{set}) \mathcal{D} = \{\mathbf{x}, \mathbf{y}\}_{i=1, \dots, n}$$

- ▶ aprender $\mathbf{f}_{\mathbf{w}} : \mathbf{x} \longrightarrow \mathbf{y}$

- ▶ Validamos con muestras nuevas

$$\mathbf{f}_{\mathbf{w}}(\mathbf{x}_{\text{nuevo}}) = ? \mathbf{y}_{\text{nuevo}}$$

APRENDIZAJE SUPERVISADO

- ▶ n muestras de datos

$$\mathbf{x}_i \in \mathbb{R}^p; i = 1, \dots, n$$

- ▶ n etiquetas $\mathbf{y}_i \in \mathbb{R}^d; i = 1, \dots, n$

- ▶ Conjunto de datos (training set) $\mathcal{D} = \{\mathbf{x}, \mathbf{y}\}_{i=1, \dots, n}$

- ▶ aprender $\mathbf{f}_w : \mathbf{x} \longrightarrow \mathbf{y}$

- ▶ Validamos con muestras nuevas

$$\mathbf{f}_w(\mathbf{x}_{\text{nuevo}}) = ? \mathbf{y}_{\text{nuevo}}$$



Perro

APRENDIZAJE SUPERVISADO

- ▶ n muestras de datos

$$\mathbf{x}_i \in \mathbb{R}^p; i = 1, \dots, n$$

- ▶ n etiquetas $\mathbf{y}_i \in \mathbb{R}^d; i = 1, \dots, n$

- ▶ Conjunto de datos (training set) $\mathcal{D} = \{\mathbf{x}, \mathbf{y}\}_{i=1, \dots, n}$

- ▶ aprender $\mathbf{f}_w : \mathbf{x} \longrightarrow \mathbf{y}$

- ▶ Validamos con muestras nuevas

$$\mathbf{f}_w(\mathbf{x}_{\text{nuevo}}) = ? \mathbf{y}_{\text{nuevo}}$$



Perro

= \mathbf{X}

APRENDIZAJE SUPERVISADO

- ▶ n muestras de datos

$$\mathbf{x}_i \in \mathbb{R}^p; i = 1, \dots, n$$

- ▶ n etiquetas $\mathbf{y}_i \in \mathbb{R}^d; i = 1, \dots, n$

- ▶ Conjunto de datos (training set) $\mathcal{D} = \{\mathbf{x}, \mathbf{y}\}_{i=1, \dots, n}$

- ▶ aprender $\mathbf{f}_w : \mathbf{x} \longrightarrow \mathbf{y}$

- ▶ Validamos con muestras nuevas

$$\mathbf{f}_w(\mathbf{x}_{\text{nuevo}}) = ? \mathbf{y}_{\text{nuevo}}$$



Perro

= \mathbf{x}

Perro

APRENDIZAJE SUPERVISADO

- ▶ n muestras de datos
 $\mathbf{x}_i \in \mathbb{R}^p; i = 1, \dots, n$
- ▶ n etiquetas $\mathbf{y}_i \in \mathbb{R}^d; i = 1, \dots, n$
- ▶ Conjunto de datos (training set) $\mathcal{D} = \{\mathbf{x}, \mathbf{y}\}_{i=1, \dots, n}$
- ▶ aprender $\mathbf{f}_w : \mathbf{x} \longrightarrow \mathbf{y}$
- ▶ Validamos con muestras nuevas
 $\mathbf{f}_w(\mathbf{x}_{\text{nuevo}}) = ? \mathbf{y}_{\text{nuevo}}$



Perro

= \mathbf{X}

Gato

Perro

APRENDIZAJE SUPERVISADO

- ▶ n muestras de datos

$$\mathbf{x}_i \in \mathbb{R}^p; i = 1, \dots, n$$

- ▶ n etiquetas $\mathbf{y}_i \in \mathbb{R}^d; i = 1, \dots, n$

- ▶ Conjunto de datos (training set) $\mathcal{D} = \{\mathbf{x}, \mathbf{y}\}_{i=1, \dots, n}$

- ▶ aprender $\mathbf{f}_w : \mathbf{x} \longrightarrow \mathbf{y}$

- ▶ Validamos con muestras nuevas

$$\mathbf{f}_w(\mathbf{x}_{\text{nuevo}}) = ? \mathbf{y}_{\text{nuevo}}$$



Perro

$$= \mathbf{x}$$

$$\mathbf{y} = \text{Gato}$$

Perro

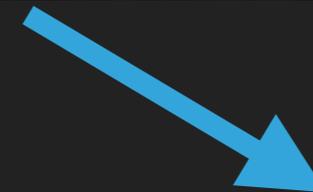
APRENDIZAJE SUPERVISADO

- ▶ n muestras de datos
 $\mathbf{x}_i \in \mathbb{R}^p; i = 1, \dots, n$
- ▶ n etiquetas $\mathbf{y}_i \in \mathbb{R}^d; i = 1, \dots, n$
- ▶ Conjunto de datos (training set) $\mathcal{D} = \{\mathbf{x}, \mathbf{y}\}_{i=1, \dots, n}$
- ▶ aprender $\mathbf{f}_w : \mathbf{x} \longrightarrow \mathbf{y}$
- ▶ Validamos con muestras nuevas
 $\mathbf{f}_w(\mathbf{x}_{\text{nuevo}}) = ? \mathbf{y}_{\text{nuevo}}$



Perro

= \mathbf{x}



$\mathbf{y} =$ Gato

Perro

APRENDIZAJE SUPERVISADO

- ▶ n muestras de datos
 $\mathbf{x}_i \in \mathbb{R}^p; i = 1, \dots, n$
- ▶ n etiquetas $\mathbf{y}_i \in \mathbb{R}^d; i = 1, \dots, n$
- ▶ Conjunto de datos (training set) $\mathcal{D} = \{\mathbf{x}, \mathbf{y}\}_{i=1, \dots, n}$
- ▶ aprender $\mathbf{f}_w : \mathbf{x} \longrightarrow \mathbf{y}$
- ▶ Validamos con muestras nuevas
 $\mathbf{f}_w(\mathbf{x}_{\text{nuevo}}) = ? \mathbf{y}_{\text{nuevo}}$



= \mathbf{x}

Perro



ALGORITMO/MODELO
 f_w

$\mathbf{y} =$ Gato Perro

APRENDIZAJE SUPERVISADO

- ▶ n muestras de datos
 $\mathbf{x}_i \in \mathbb{R}^p; i = 1, \dots, n$
- ▶ n etiquetas $\mathbf{y}_i \in \mathbb{R}^d; i = 1, \dots, n$
- ▶ Conjunto de datos (training set) $\mathcal{D} = \{\mathbf{x}, \mathbf{y}\}_{i=1, \dots, n}$
- ▶ aprender $\mathbf{f}_w : \mathbf{x} \longrightarrow \mathbf{y}$
- ▶ Validamos con muestras nuevas
 $\mathbf{f}_w(\mathbf{x}_{\text{nuevo}}) = ? \mathbf{y}_{\text{nuevo}}$



= \mathbf{x}

Perro

ALGORITMO/MODELO
 f_w

$\mathbf{y} =$ Gato

Perro

APRENDIZAJE SUPERVISADO

- ▶ n muestras de datos
 $\mathbf{x}_i \in \mathbb{R}^p; i = 1, \dots, n$
- ▶ n etiquetas $\mathbf{y}_i \in \mathbb{R}^d; i = 1, \dots, n$
- ▶ Conjunto de datos (training set) $\mathcal{D} = \{\mathbf{x}, \mathbf{y}\}_{i=1, \dots, n}$
- ▶ aprender $\mathbf{f}_w : \mathbf{x} \longrightarrow \mathbf{y}$
- ▶ Validamos con muestras nuevas
 $\mathbf{f}_w(\mathbf{x}_{\text{nuevo}}) = ? \mathbf{y}_{\text{nuevo}}$



= \mathbf{x}

Perro



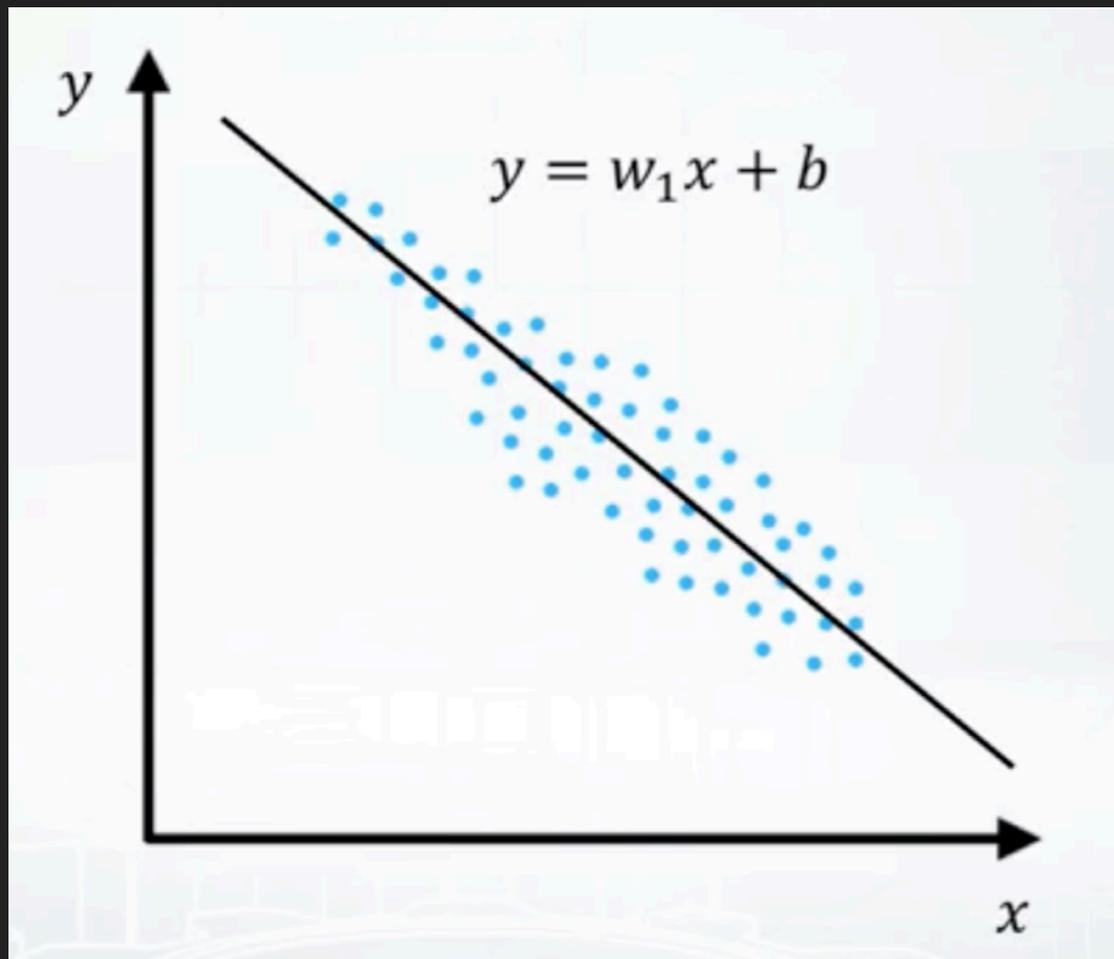
$\mathbf{y} =$ Gato

Perro

APRENDIZAJE SUPERVISADO

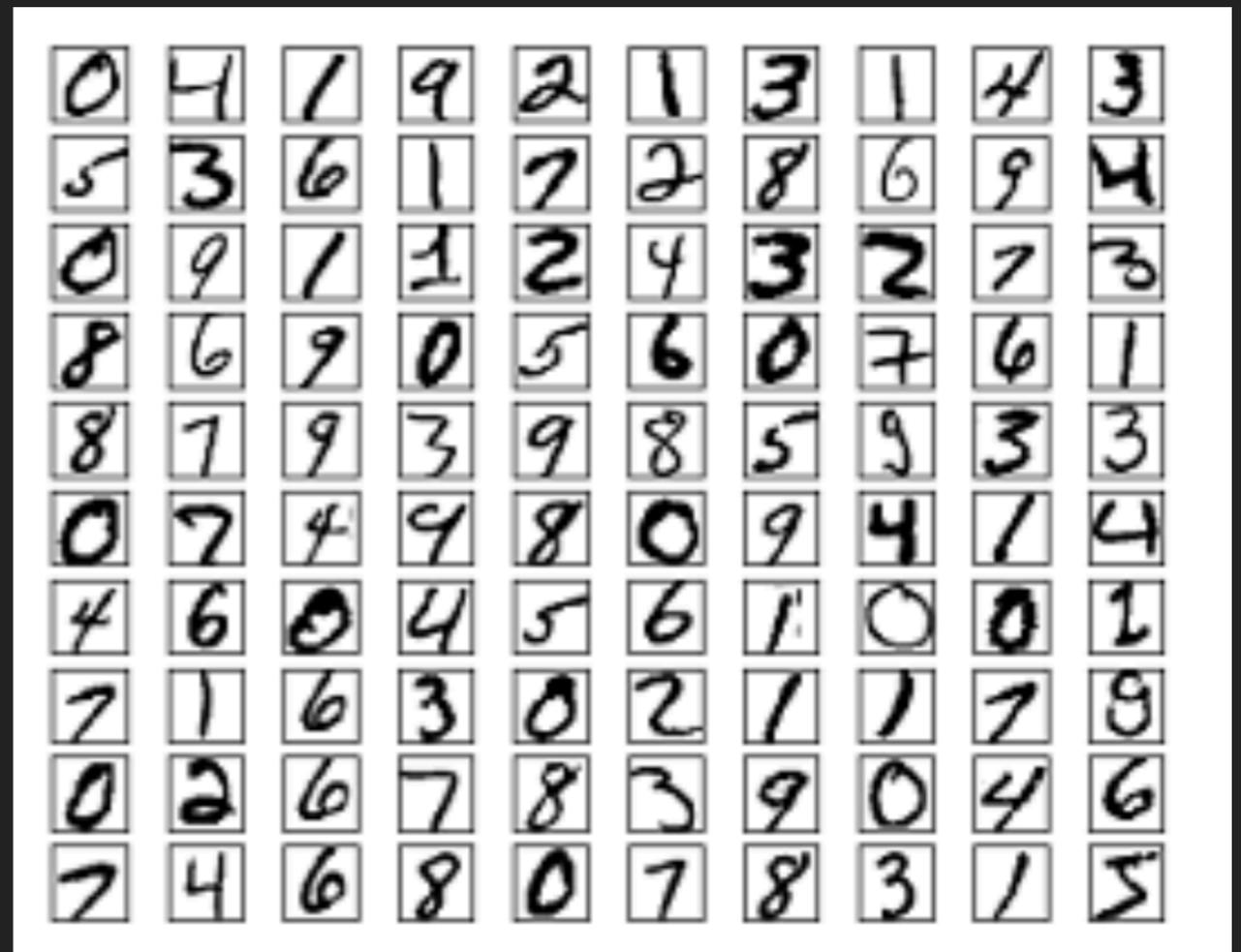
▶ Regresión

"etiquetas" y: continuo



▶ Clasificación

"etiquetas" y: valores discretos



DESCENSO DE GRADIENTE GRADIENT DESCENT

<https://github.com/pvigier/gradient-descent/blob/master/README.md>

DESCENSO DE GRADIENTE GRADIENT DESCENT

- ▶ Función de pérdida para cada i

DESCENSO DE GRADIENTE GRADIENT DESCENT

- ▶ Función de pérdida para cada i

$$\mathcal{L} = \mathcal{L}[f_w(x_i), y_i] = ||f_w(x_i) - y_i||^2$$

DESCENSO DE GRADIENTE GRADIENT DESCENT

- ▶ Función de pérdida para cada i

$$\mathcal{L} = \mathcal{L}[f_w(x_i), y_i] = ||f_w(x_i) - y_i||^2$$

- ▶ Riesgo Empirico $\mathcal{R}(f_w) = \frac{1}{n} \sum_{i=1}^n \mathcal{L}$

DESCENSO DE GRADIENTE GRADIENT DESCENT

- ▶ Función de pérdida para cada i

$$\mathcal{L} = \mathcal{L}[f_w(x_i), y_i] = ||f_w(x_i) - y_i||^2$$

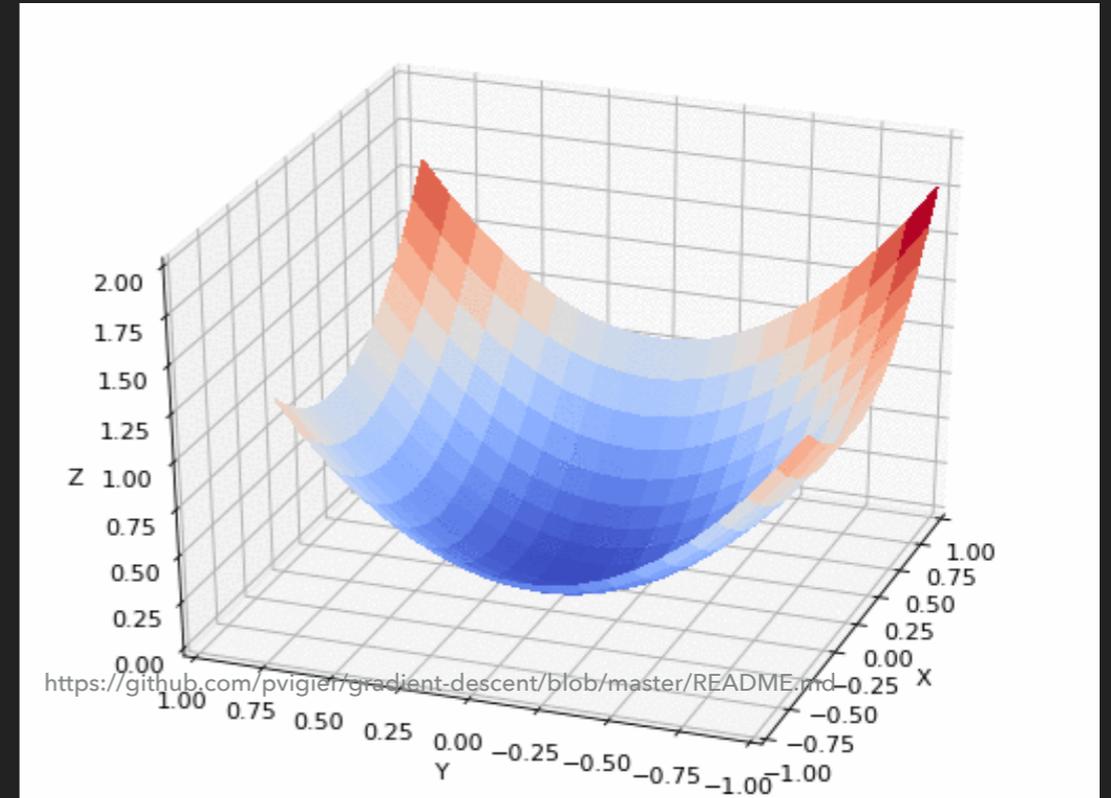
- ▶ Riesgo Empirico $\mathcal{R}(f_w) = \frac{1}{n} \sum_{i=1}^n \mathcal{L}$

DESCENSO DE GRADIENTE GRADIENT DESCENT

- ▶ Función de pérdida para cada i

$$\mathcal{L} = \mathcal{L}[f_w(x_i), y_i] = ||f_w(x_i) - y_i||^2$$

- ▶ Riesgo Empirico $\mathcal{R}(f_w) = \frac{1}{n} \sum_{i=1}^n \mathcal{L}$



DESCENSO DE GRADIENTE GRADIENT DESCENT

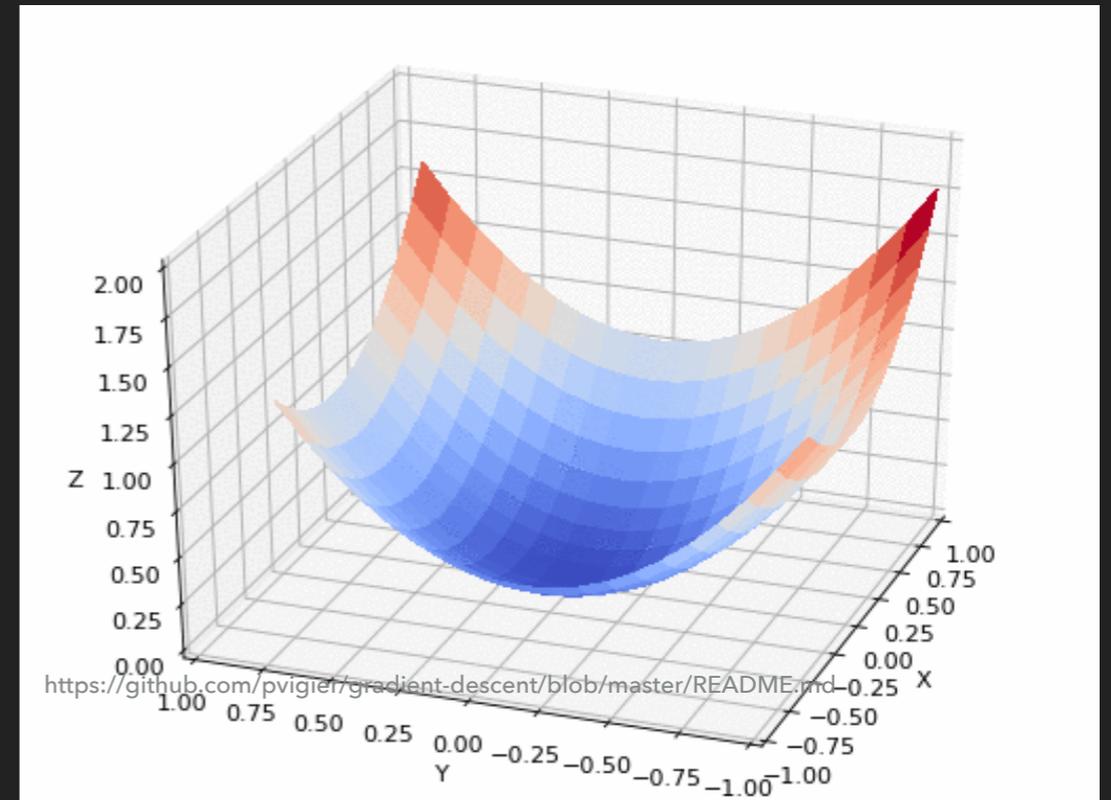
- ▶ Función de pérdida para cada i

$$\mathcal{L} = \mathcal{L}[f_w(x_i), y_i] = ||f_w(x_i) - y_i||^2$$

- ▶ Riesgo Empirico $\mathcal{R}(f_w) = \frac{1}{n} \sum_{i=1}^n \mathcal{L}$

- ▶ Gradiente descendiente

$$w \rightarrow w - \lambda \nabla \mathcal{R}(f_w)$$



DESCENSO DE GRADIENTE GRADIENT DESCENT

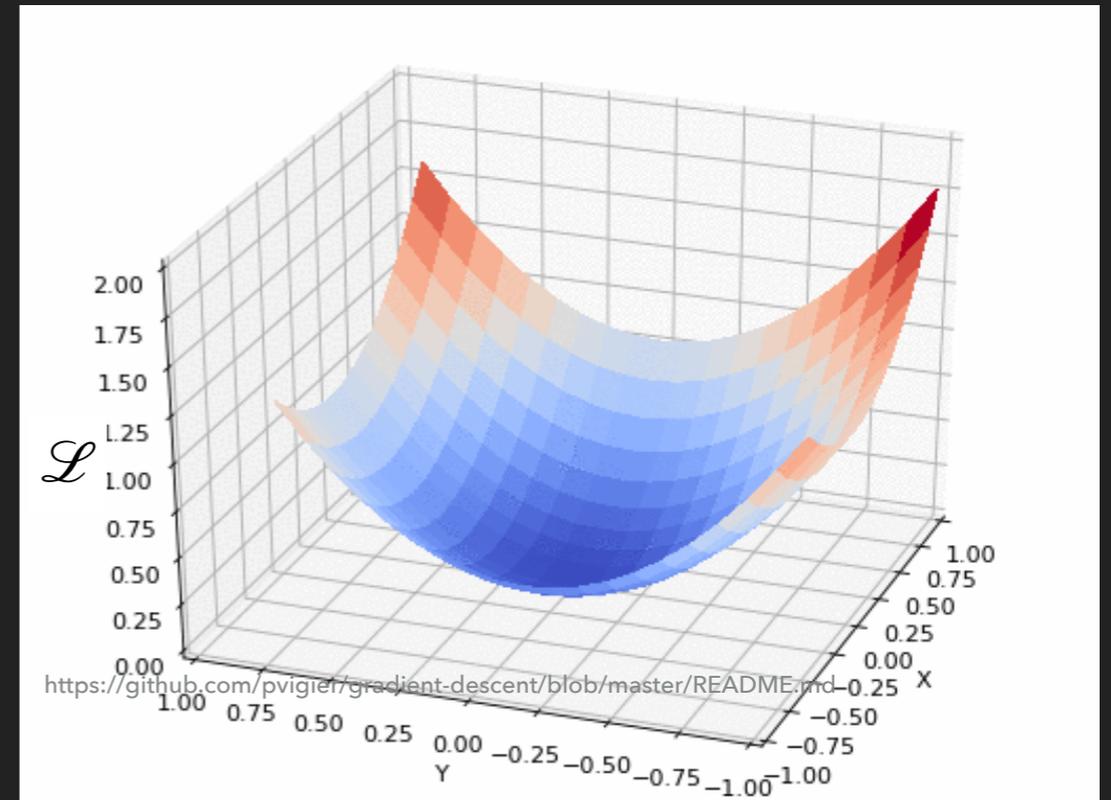
- ▶ Función de pérdida para cada i

$$\mathcal{L} = \mathcal{L}[f_w(x_i), y_i] = ||f_w(x_i) - y_i||^2$$

- ▶ Riesgo Empirico $\mathcal{R}(f_w) = \frac{1}{n} \sum_{i=1}^n \mathcal{L}$

- ▶ Gradiente descendiente

$$w \rightarrow w - \lambda \nabla \mathcal{R}(f_w)$$

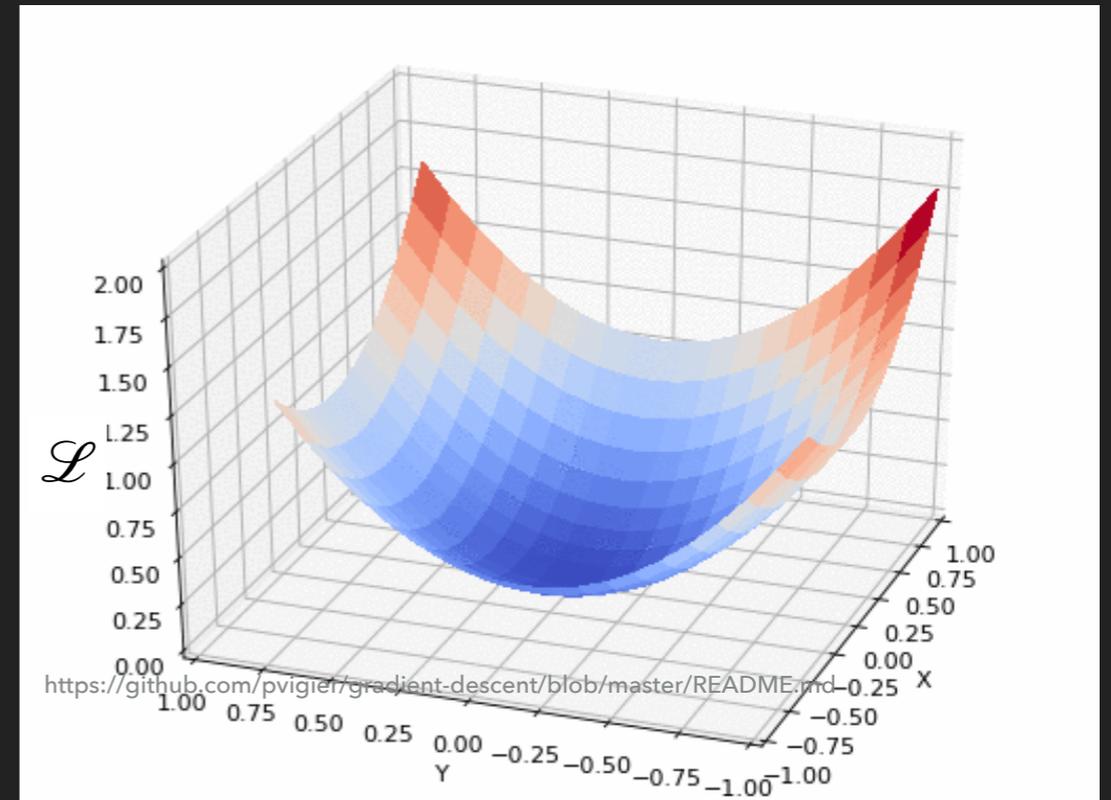


DESCENSO DE GRADIENTE GRADIENT DESCENT

- ▶ Función de pérdida para cada i

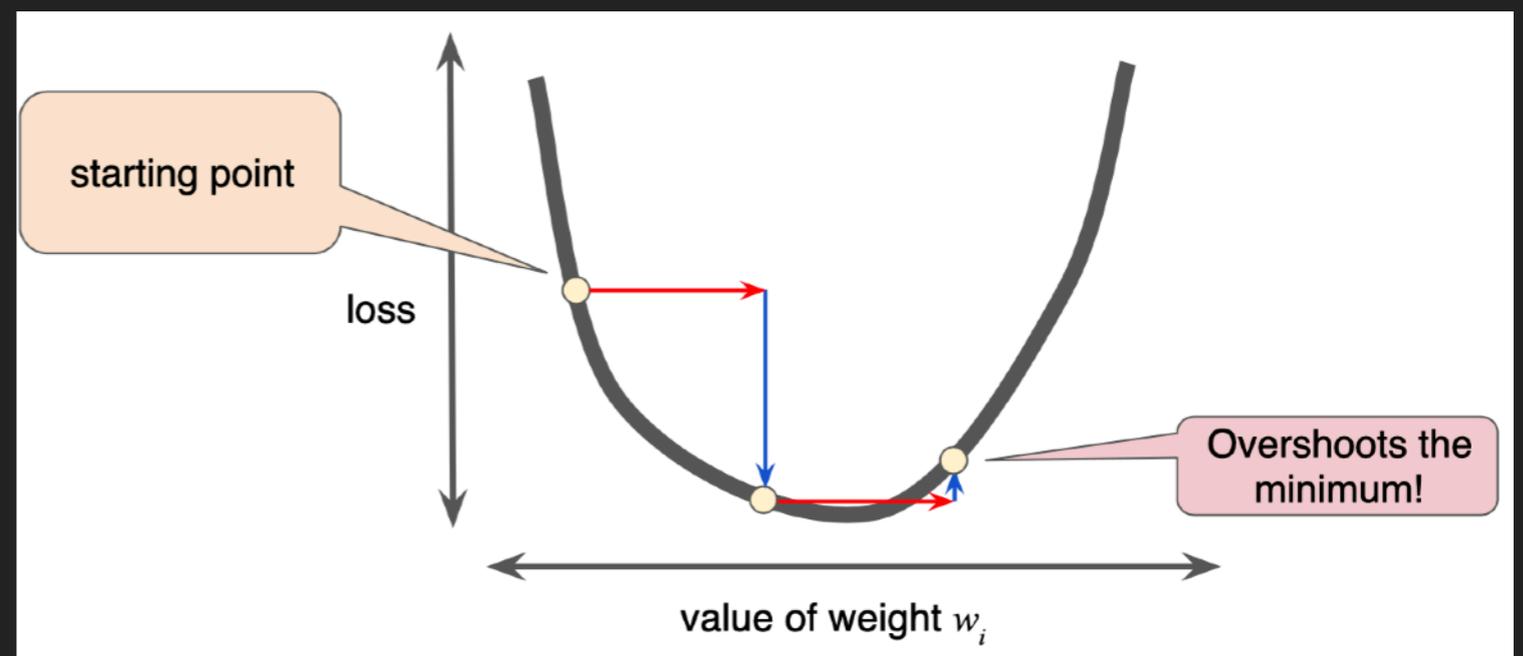
$$\mathcal{L} = \mathcal{L}[f_w(x_i), y_i] = ||f_w(x_i) - y_i||^2$$

- ▶ Riesgo Empirico $\mathcal{R}(f_w) = \frac{1}{n} \sum_{i=1}^n \mathcal{L}$



- ▶ Gradiente descendiente

$$w \rightarrow w - \lambda \nabla \mathcal{R}(f_w)$$



DESCENSO DE GRADIENTE GRADIENT DESCENT

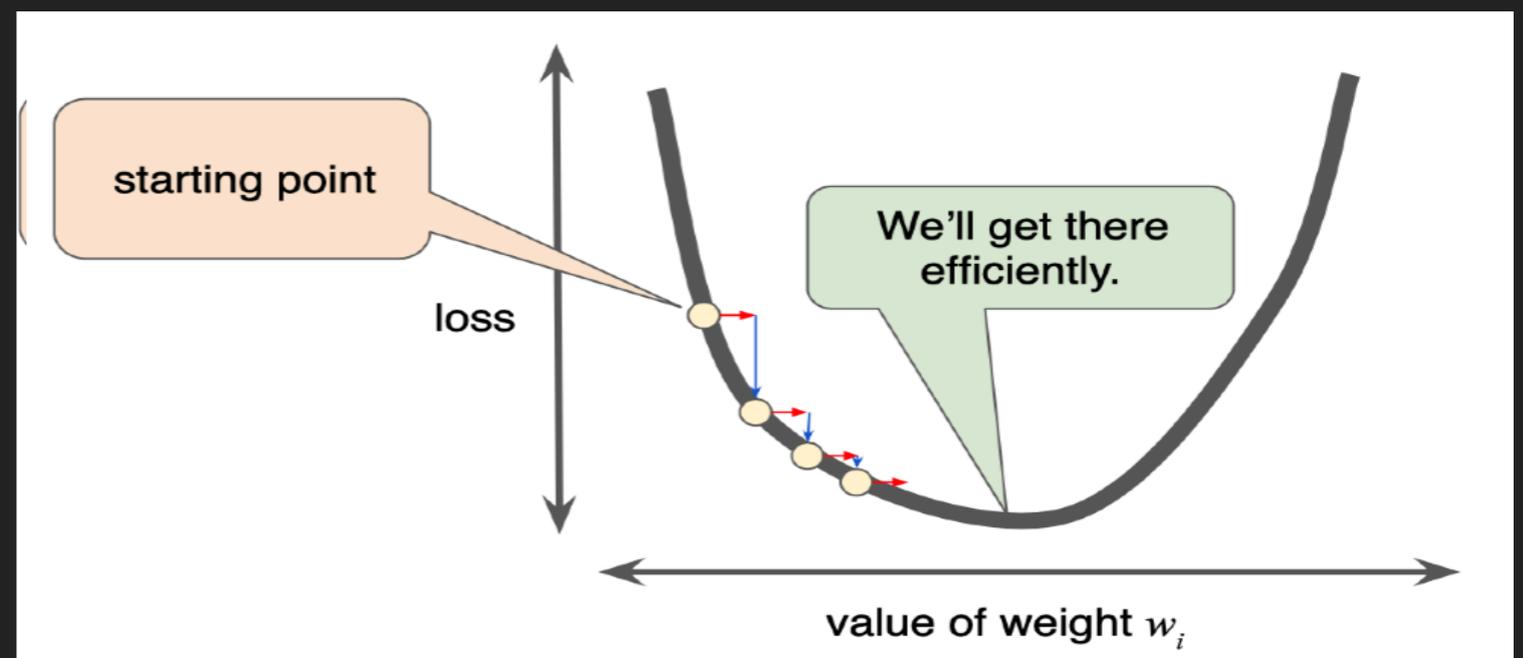
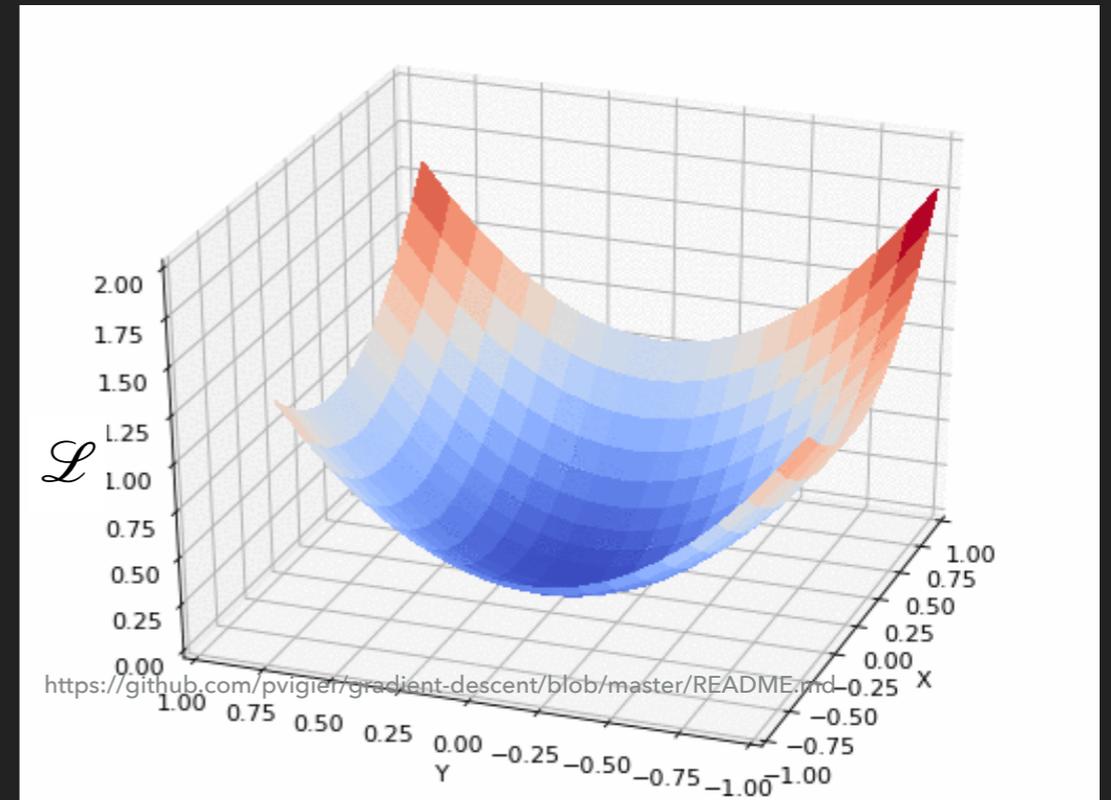
- ▶ Función de pérdida para cada i

$$\mathcal{L} = \mathcal{L}[f_w(x_i), y_i] = ||f_w(x_i) - y_i||^2$$

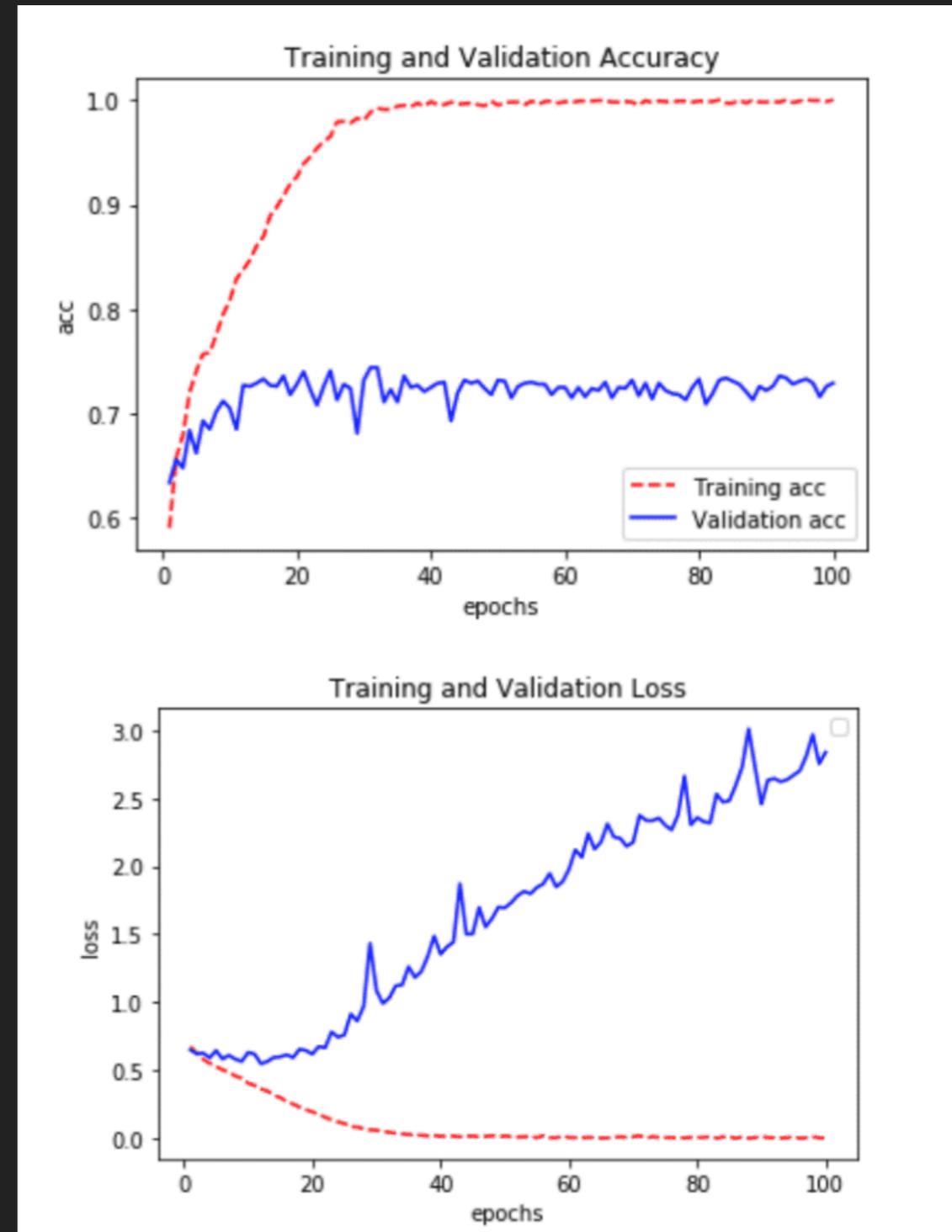
- ▶ Riesgo Empirico $\mathcal{R}(f_w) = \frac{1}{n} \sum_{i=1}^n \mathcal{L}$

- ▶ Gradiente descendiente

$$w \rightarrow w - \lambda \nabla \mathcal{R}(f_w)$$

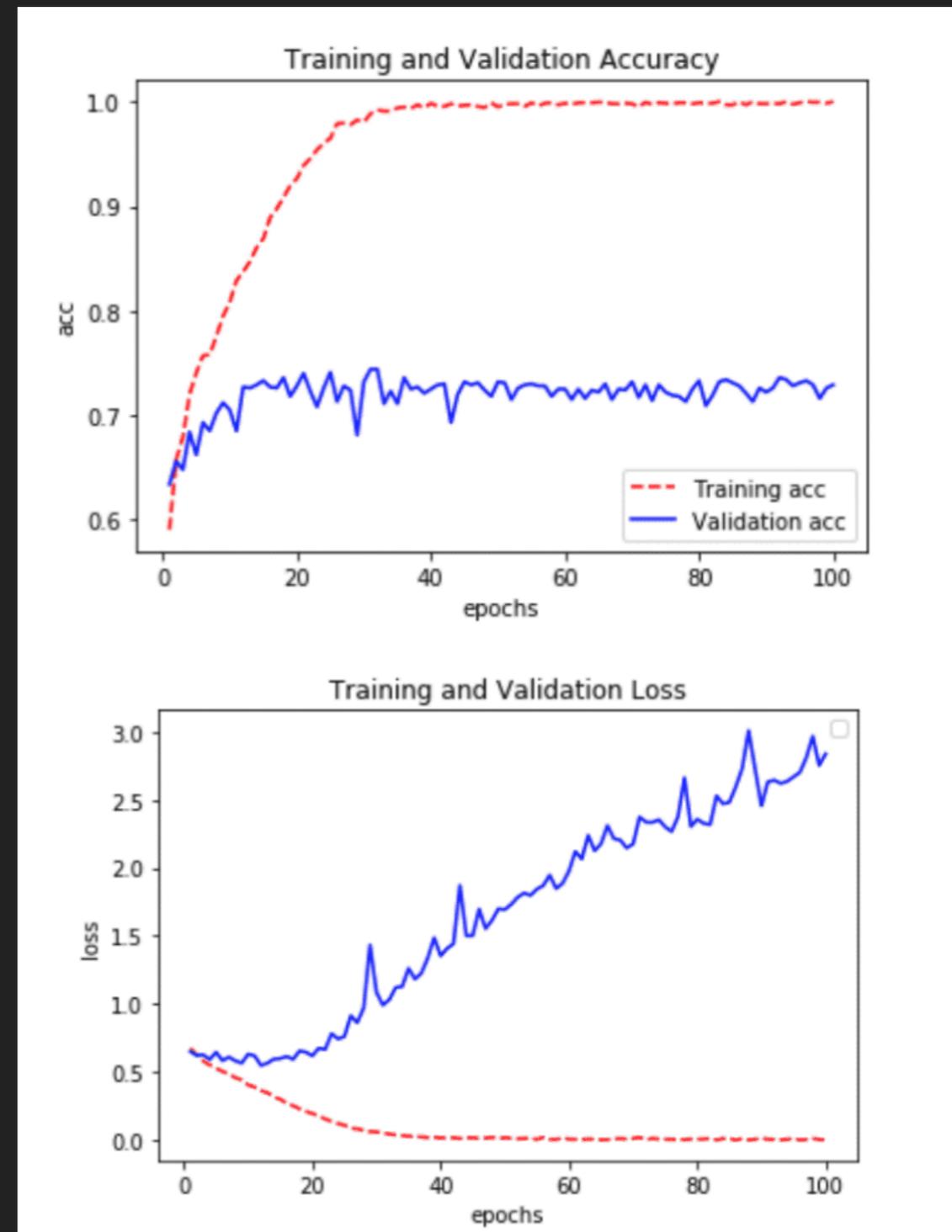


SOBRE-AJUSTE



SOBRE-AJUSTE

- ▶ Conjunto de Validación = no se ajustan los parámetros w
- ▶ Validación Cruzada
 - ▶ requiere entrenar modelo k veces para k -fold VC
 - ▶ No muy eficiente para modelos de Deep Learning



SOBRE-AJUSTE

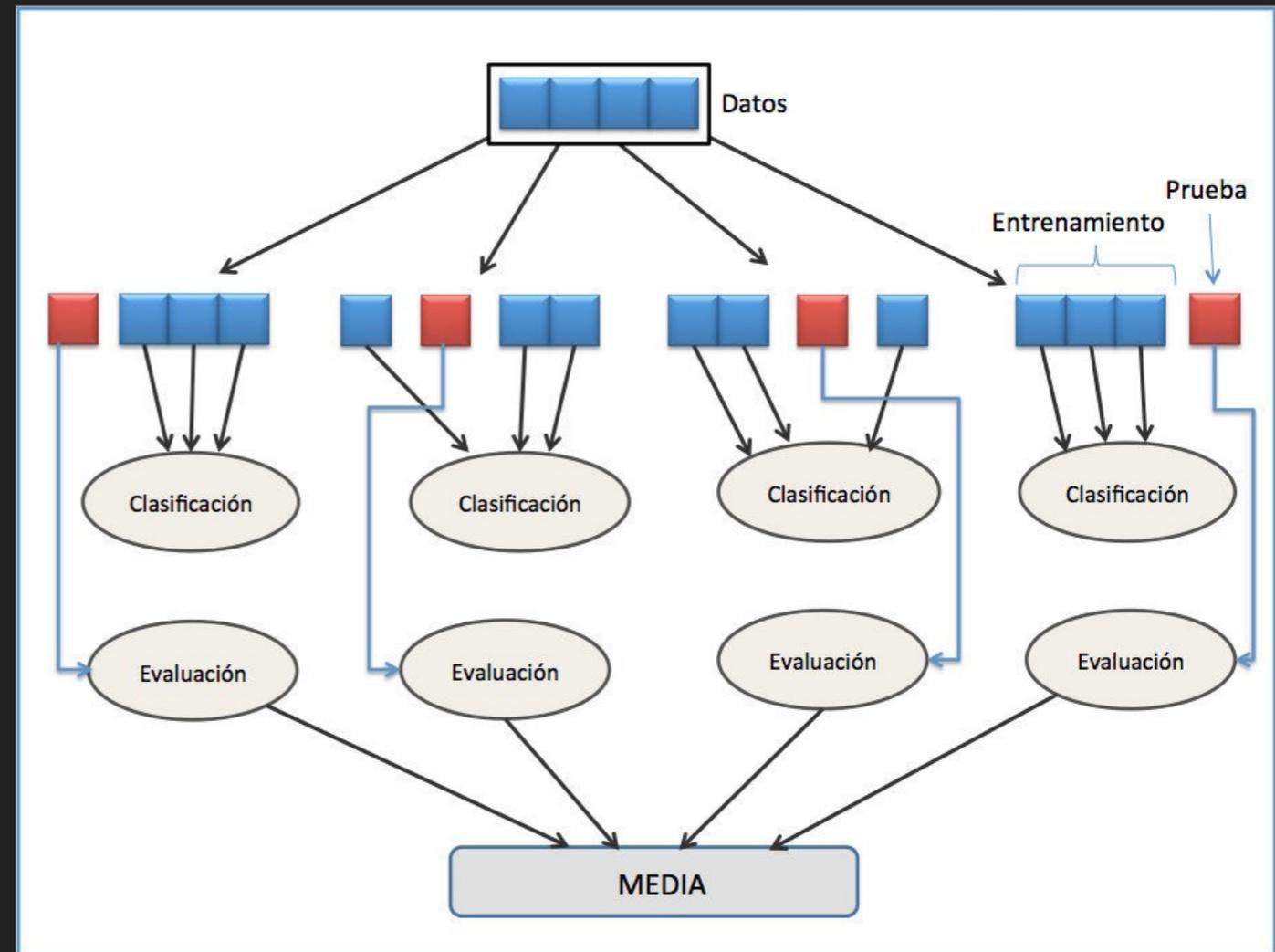
- ▶ Conjunto de Validación = no se ajustan los parámetros w
- ▶ Validación Cruzada
 - ▶ requiere entrenar modelo k veces para k -fold VC
 - ▶ No muy eficiente para modelos de Deep Learning

SOBRE-AJUSTE

- ▶ Conjunto de Validación = no se ajustan los parámetros w
- ▶ Validación Cruzada
 - ▶ requiere entrenar modelo k veces para k -fold VC
 - ▶ No muy eficiente para modelos de Deep Learning

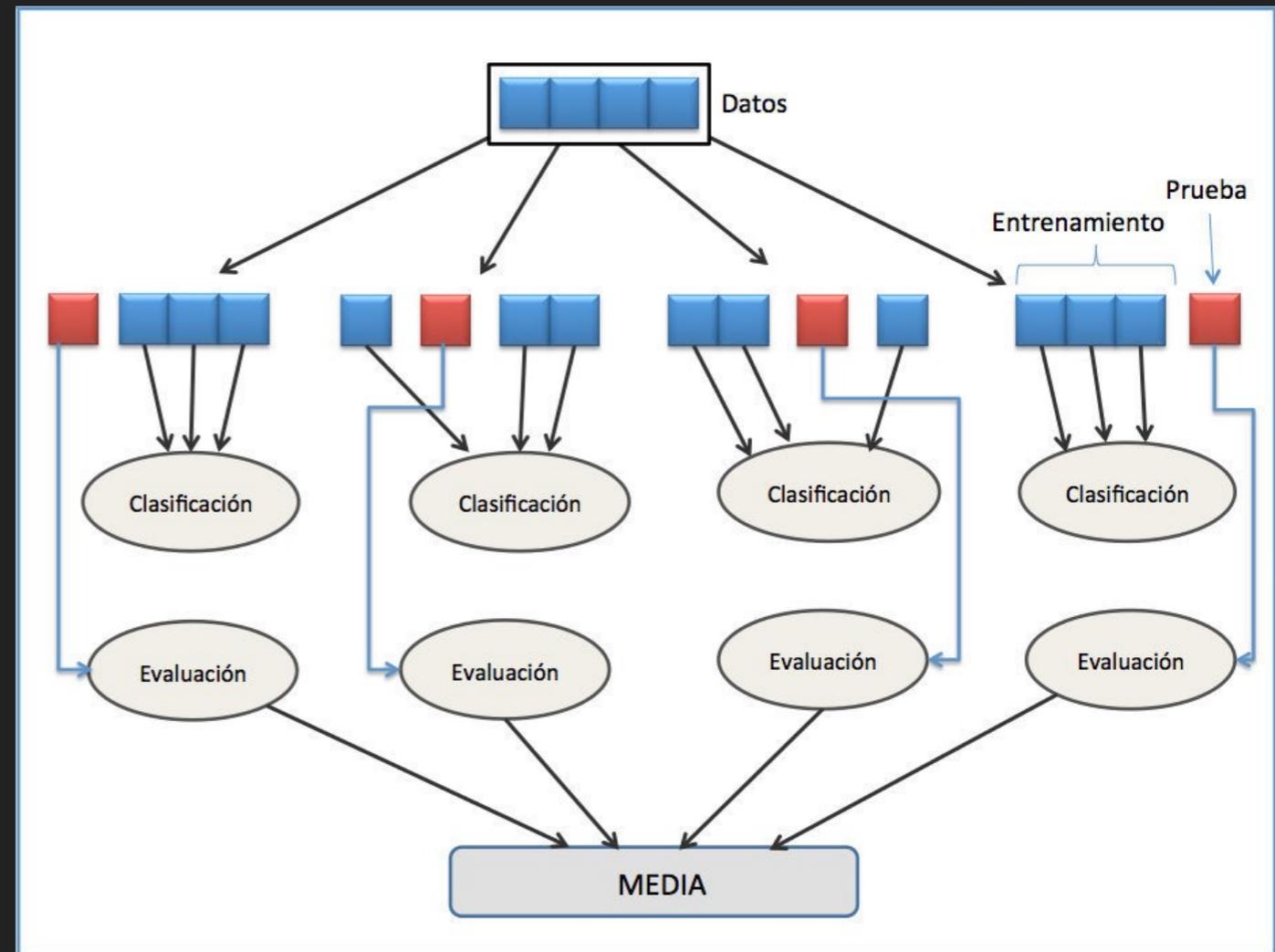
SOBRE-AJUSTE

- ▶ Conjunto de Validación = no se ajustan los parámetros w
- ▶ Validación Cruzada
 - ▶ requiere entrenar modelo k veces para k -fold VC
 - ▶ No muy eficiente para modelos de Deep Learning



SOBRE-AJUSTE

- ▶ Conjunto de Validación = no se ajustan los parámetros w
- ▶ Validación Cruzada
 - ▶ requiere entrenar modelo k veces para k -fold VC
 - ▶ No muy eficiente para modelos de Deep Learning



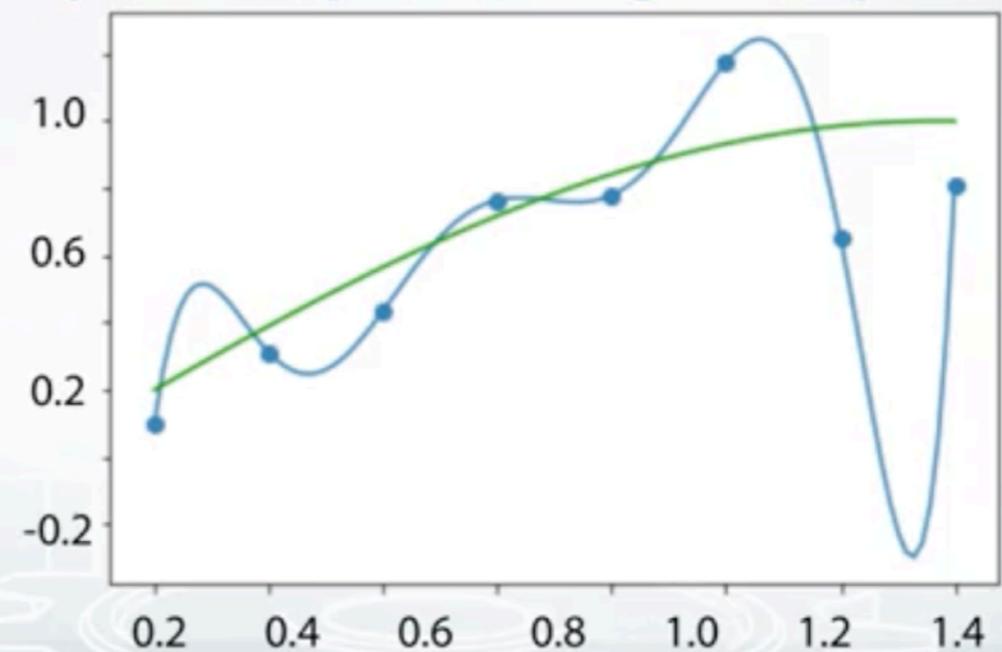
REGULARIZACIÓN

Training set: $\{0.2, 0.4, \dots, 1.6\}$, $y = \sin(x) + \epsilon$

Model: $a(x) = b + w_1x + w_2x^2 + \dots + w_8x^8$

Parameters: $(130.0, -525.8, \dots, 102.6)$

Model just incorporates target into parameters!



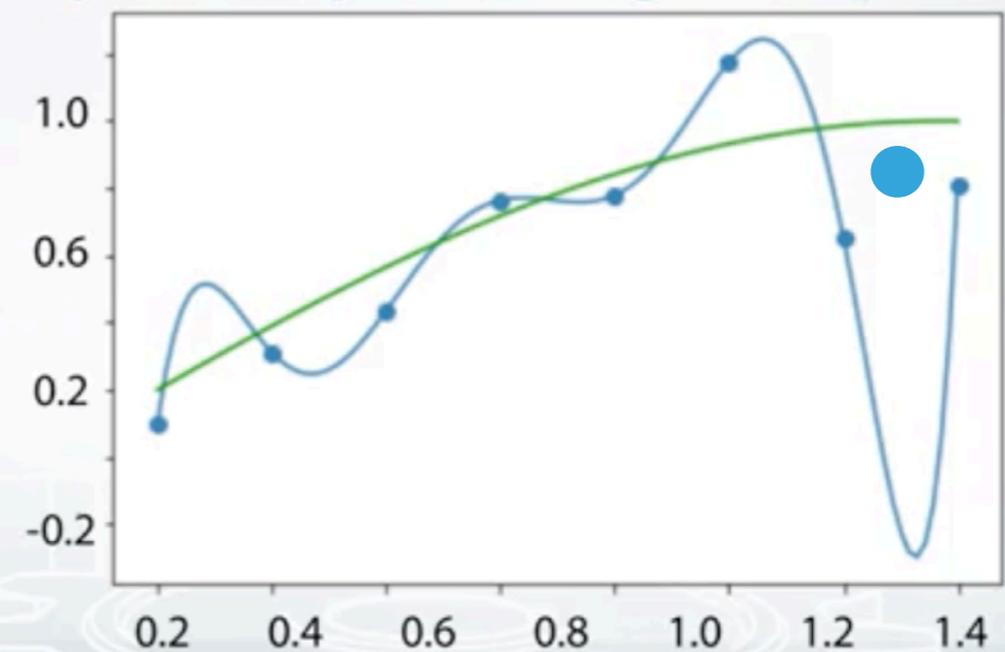
REGULARIZACIÓN

Training set: $\{0.2, 0.4, \dots, 1.6\}$, $y = \sin(x) + \epsilon$

Model: $a(x) = b + w_1x + w_2x^2 + \dots + w_8x^8$

Parameters: $(130.0, -525.8, \dots, 102.6)$

Model just incorporates target into parameters!



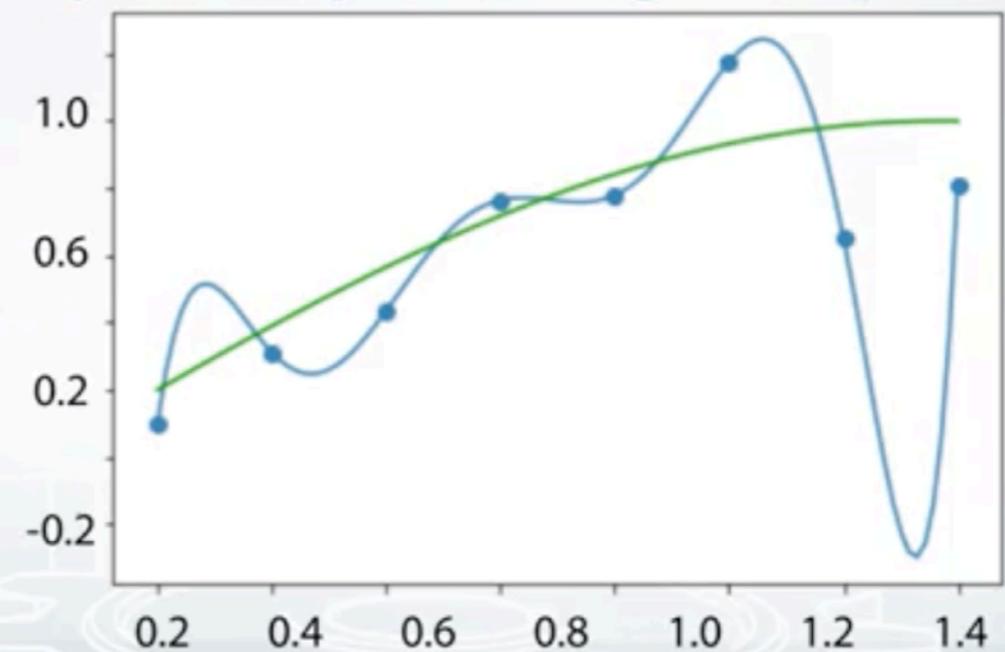
REGULARIZACIÓN

Training set: $\{0.2, 0.4, \dots, 1.6\}$, $y = \sin(x) + \epsilon$

Model: $a(x) = b + w_1x + w_2x^2 + \dots + w_8x^8$

Parameters: $(130.0, -525.8, \dots, 102.6)$

Model just incorporates target into parameters!



REGULARIZACIÓN

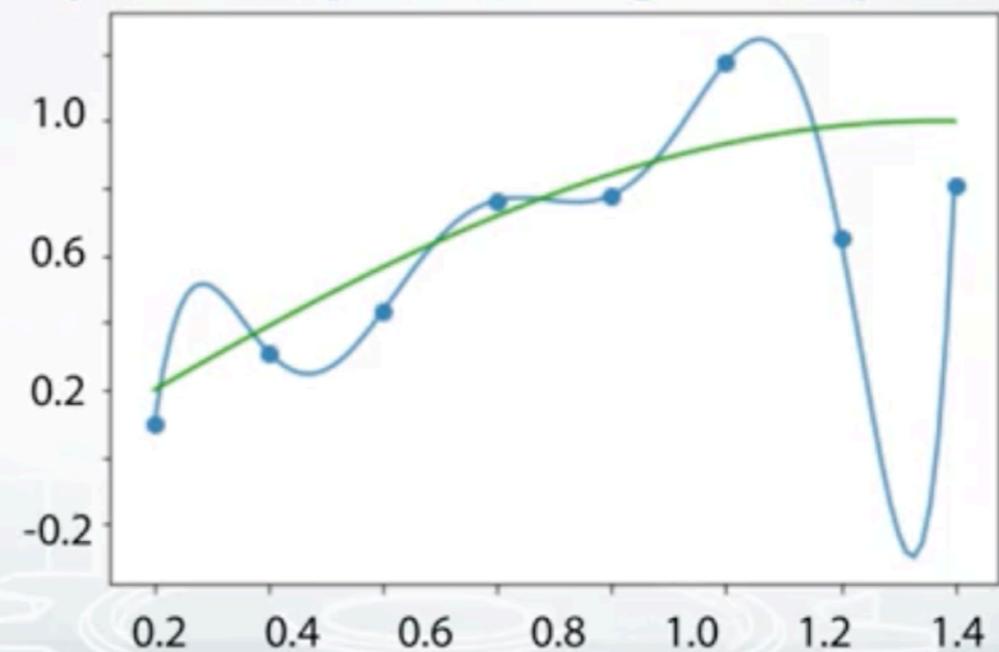
► Penalización L2

Training set: $\{0.2, 0.4, \dots, 1.6\}$, $y = \sin(x) + \epsilon$

Model: $a(x) = b + w_1x + w_2x^2 + \dots + w_8x^8$

Parameters: $(130.0, -525.8, \dots, 102.6)$

Model just incorporates target into parameters!



REGULARIZACIÓN

▶ Penalización L2

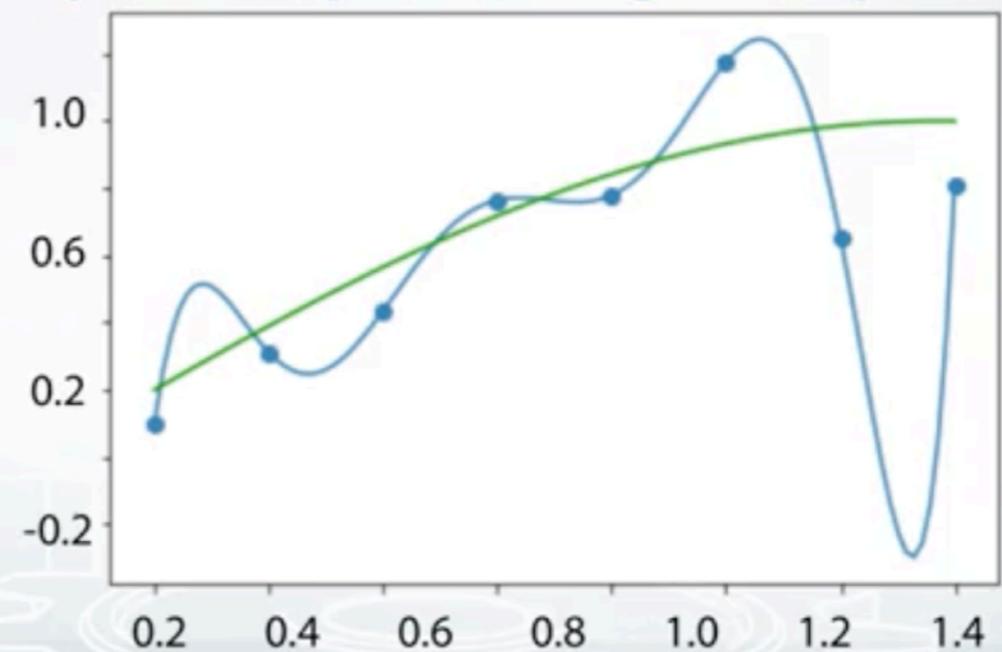
$$\mathcal{L}_{L2} = \mathcal{L} + \lambda ||\mathbf{w}'||^2$$

Training set: $\{0.2, 0.4, \dots, 1.6\}$, $y = \sin(x) + \epsilon$

Model: $a(x) = b + w_1x + w_2x^2 + \dots + w_8x^8$

Parameters: (130.0, -525.8, ..., 102.6)

Model just incorporates target into parameters!



REGULARIZACIÓN

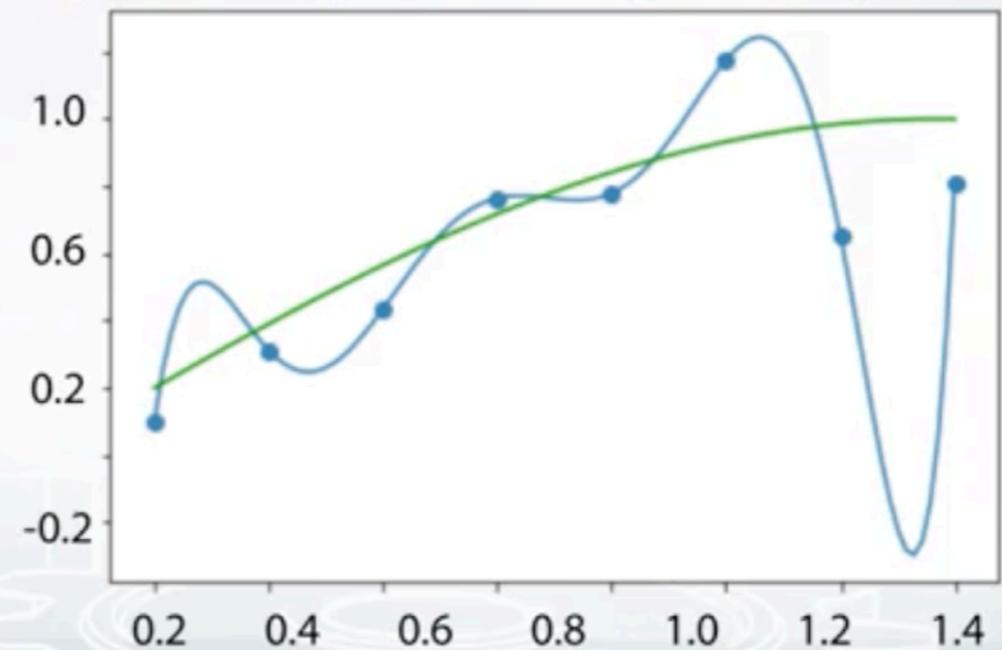
- ▶ Penalización L2
- ▶ $\mathcal{L}_{L2} = \mathcal{L} + \lambda ||\mathbf{w}'||^2$
- ▶ conduce parametros \mathbf{w} a 0

Training set: $\{0.2, 0.4, \dots, 1.6\}$, $y = \sin(x) + \epsilon$

Model: $a(x) = b + w_1x + w_2x^2 + \dots + w_8x^8$

Parameters: (130.0, -525.8, ..., 102.6)

Model just incorporates target into parameters!



REGULARIZACIÓN

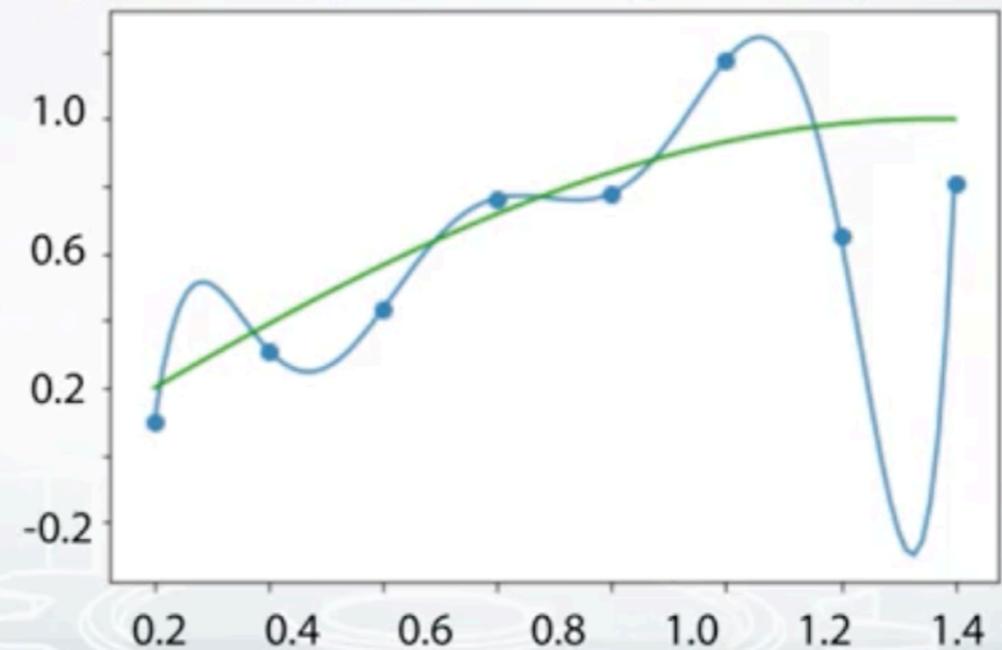
- ▶ Penalización L2
- ▶ $\mathcal{L}_{L2} = \mathcal{L} + \lambda ||\mathbf{w}'||^2$
- ▶ conduce parametros \mathbf{w} a 0
- ▶ Puede ser optimizada con métodos de descenso de gradiente

Training set: $\{0.2, 0.4, \dots, 1.6\}$, $y = \sin(x) + \epsilon$

Model: $a(x) = b + w_1x + w_2x^2 + \dots + w_8x^8$

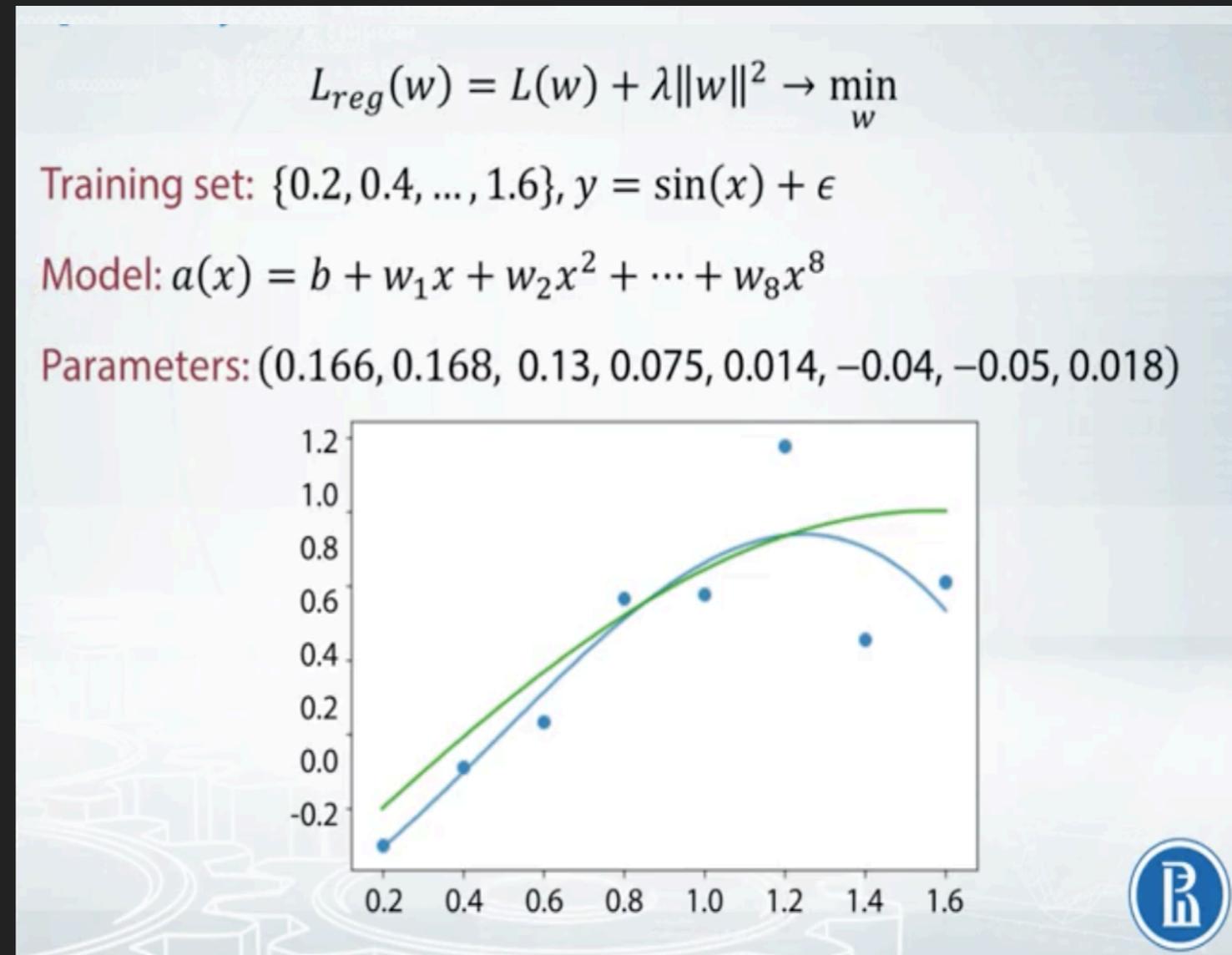
Parameters: (130.0, -525.8, ..., 102.6)

Model just incorporates target into parameters!



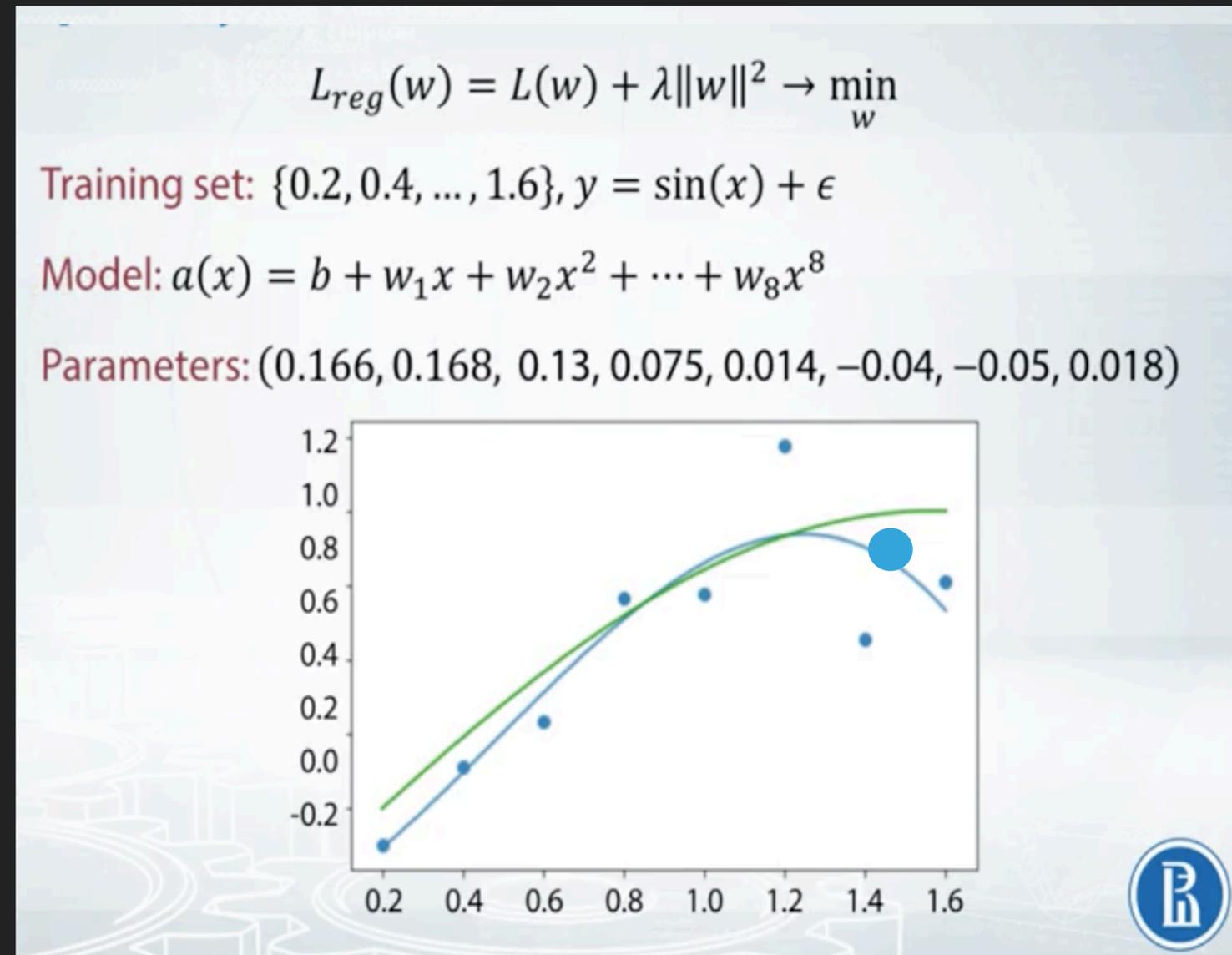
REGULARIZACIÓN

- ▶ Penalización L2
- ▶ $\mathcal{L}_{L2} = \mathcal{L} + \lambda ||\mathbf{w}'||^2$
- ▶ conduce parametros \mathbf{w} a 0
- ▶ Puede ser optimizada con métodos de descenso de gradiente



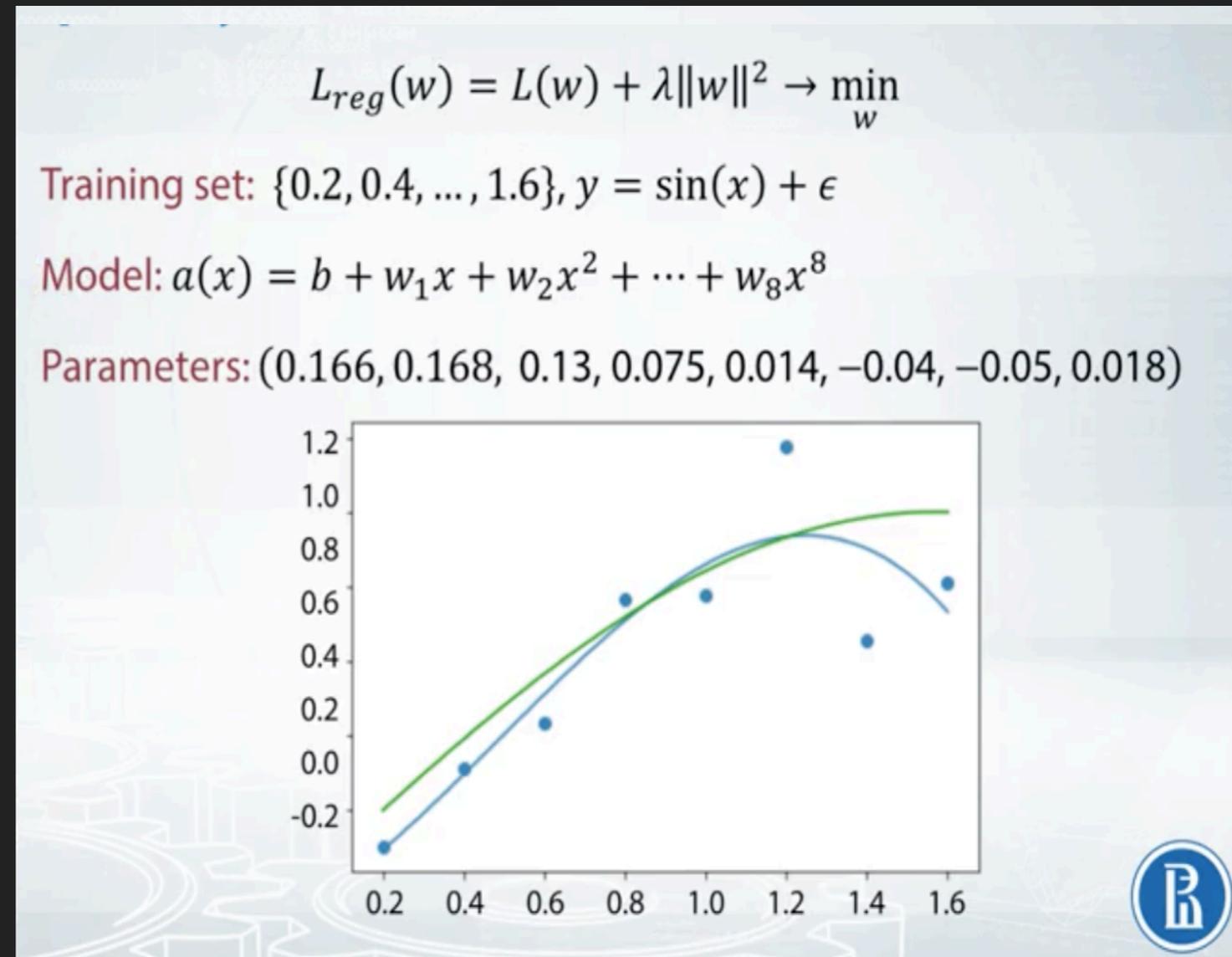
REGULARIZACIÓN

- ▶ Penalización L2
- ▶ $\mathcal{L}_{L2} = \mathcal{L} + \lambda ||\mathbf{w}'||^2$
- ▶ conduce parametros \mathbf{w} a 0
- ▶ Puede ser optimizada con métodos de descenso de gradiente



REGULARIZACIÓN

- ▶ Penalización L2
- ▶ $\mathcal{L}_{L2} = \mathcal{L} + \lambda ||\mathbf{w}'||^2$
- ▶ conduce parametros \mathbf{w} a 0
- ▶ Puede ser optimizada con métodos de descenso de gradiente



TAREAS DE APRENDIZAJE SUPERVISADO EN HEP

- ▶ Identificación/clasificación de partículas
- ▶ Identificación de señales sobre background (nueva física)
- ▶ Rapido y eficiente rastreo de partículas
- ▶ Clasificación de jets
- ▶ Triggering
- ▶ Procesamiento de datos

APRENDIZAJE DE MAQUINAS

APRENDIZAJE NO SUPERVISADO

APRENDIZAJE NO SUPERVISADO

- ▶ Set de datos $\mathcal{D} = \{\mathbf{x}\}_{i=1,\dots,n}$

APRENDIZAJE NO SUPERVISADO

- ▶ Set de datos $\mathcal{D} = \{\mathbf{x}\}_{i=1,\dots,n}$
- ▶ Recuperar estructura subyacente

APRENDIZAJE NO SUPERVISADO

- ▶ Set de datos $\mathcal{D} = \{\mathbf{x}\}_{i=1,\dots,n}$
- ▶ Recuperar estructura subyacente
- ▶ - reducción dimensional, clustering,
modelos generativos

APRENDIZAJE NO SUPERVISADO

- ▶ Set de datos $\mathcal{D} = \{\mathbf{x}\}_{i=1,\dots,n}$
- ▶ Recuperar estructura subyacente
- ▶ - reducción dimensional, clustering,
modelos generativos
- ▶ **e.g.**

APRENDIZAJE NO SUPERVISADO

- ▶ Set de datos $\mathcal{D} = \{\mathbf{x}\}_{i=1,\dots,n}$
- ▶ Recuperar estructura subyacente
- ▶ - reducción dimensional, clustering,
modelos generativos
- ▶ **e.g.**
 - ▶ aprender/reconstruir $P(\mathbf{x})$

APRENDIZAJE NO SUPERVISADO

- ▶ Set de datos $\mathcal{D} = \{\mathbf{x}\}_{i=1,\dots,n}$
- ▶ Recuperar estructura subyacente
- ▶ - reducción dimensional, clustering, **modelos generativos**
- ▶ **e.g.**
 - ▶ aprender/reconstruir $P(\mathbf{x})$
 - ▶ Maximizando $\mathcal{L} = P(\theta | x)$

APRENDIZAJE NO SUPERVISADO

- ▶ Set de datos $\mathcal{D} = \{\mathbf{x}\}_{i=1,\dots,n}$
- ▶ Recuperar estructura subyacente
- ▶ - reducción dimensional, clustering, **modelos generativos**
- ▶ **e.g.**
 - ▶ aprender/reconstruir $P(\mathbf{x})$
 - ▶ Maximizando $\mathcal{L} = P(\theta | x)$
 - ▶ muestrear nuevos ejemplos

APRENDIZAJE NO SUPERVISADO

- ▶ Set de datos $\mathcal{D} = \{\mathbf{x}\}_{i=1,\dots,n}$
- ▶ Recuperar estructura subyacente
- ▶ - reducción dimensional, clustering, **modelos generativos**
- ▶ **e.g.**
 - ▶ aprender/reconstruir $P(\mathbf{x})$
 - ▶ Maximizando $\mathcal{L} = P(\theta | x)$
 - ▶ muestrear nuevos ejemplos

APRENDIZAJE NO SUPERVISADO

- ▶ Set de datos $\mathcal{D} = \{\mathbf{x}\}_{i=1,\dots,n}$
- ▶ Recuperar estructura subyacente
- ▶ - reducción dimensional, clustering, **modelos generativos**
- ▶ **e.g.**
 - ▶ aprender/reconstruir $P(\mathbf{x})$
 - ▶ Maximizando $\mathcal{L} = P(\theta | x)$
 - ▶ muestrear nuevos ejemplos

GAN (Red Generativa Antagonica)

APRENDIZAJE NO SUPERVISADO

- ▶ Set de datos $\mathcal{D} = \{\mathbf{x}\}_{i=1,\dots,n}$
- ▶ Recuperar estructura subyacente
- ▶ - reducción dimensional, clustering,
modelos generativos
- ▶ **e.g.**
 - ▶ aprender/reconstruir $P(\mathbf{x})$
 - ▶ Maximizando $\mathcal{L} = P(\theta | x)$
 - ▶ muestrear nuevos ejemplos

GAN (Red Generativa Antagonica)

<https://www.thispersondoesnotexist.com>

APRENDIZAJE NO SUPERVISADO

- ▶ Set de datos $\mathcal{D} = \{\mathbf{x}\}_{i=1,\dots,n}$
- ▶ Recuperar estructura subyacente
- ▶ - reducción dimensional, clustering, **modelos generativos**
- ▶ **e.g.**
 - ▶ aprender/reconstruir $P(\mathbf{x})$
 - ▶ Maximizando $\mathcal{L} = P(\theta | x)$
 - ▶ muestrear nuevos ejemplos

GAN (Red Generativa Antagonica)

<https://www.thispersondoesnotexist.com>



REDUCCIÓN DIMENSIONAL

- ▶ Proyectar o “encajar” datos en espacio dimensional menor al original



DESAFÍOS DE DATOS EN DIMENSIONES ALTAS

DESAFÍOS DE DATOS EN DIMENSIONES ALTAS

- ▶ Vive cerca del borde del espacio de muestra
hipercubo $C = [-e/2, e/2]^D$

DESAFÍOS DE DATOS EN DIMENSIONES ALTAS

- ▶ Vive cerca del borde del espacio de muestra

hipercubo $C = [-e/2, e/2]^D$

hiperesfera $S = \{ ||\mathbf{x}|| > e/2 \}$

DESAFÍOS DE DATOS EN DIMENSIONES ALTAS

- ▶ Vive cerca del borde del espacio de muestra
hipercubo $C = [-e/2, e/2]^D$

hiperesfera $S = \{ ||\mathbf{x}|| > e/2 \}$

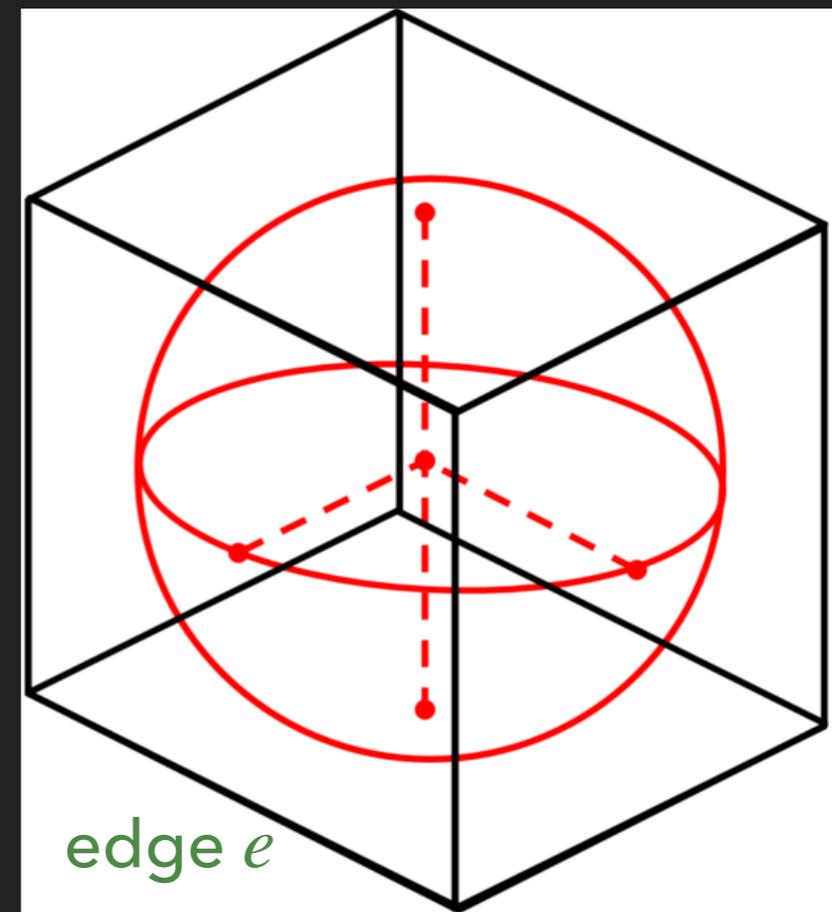
$$P(x \in S | x \in C) = \frac{\text{Vol } S}{\text{Vol } C} \sim \left(\frac{1}{2}\right)^D$$

DESAFÍOS DE DATOS EN DIMENSIONES ALTAS

- ▶ Vive cerca del borde del espacio de muestra
hipercubo $C = [-e/2, e/2]^D$

hiperesfera $S = \{ ||\mathbf{x}|| > e/2 \}$

$$P(x \in S | x \in C) = \frac{\text{Vol } S}{\text{Vol } C} \sim \left(\frac{1}{2}\right)^D$$



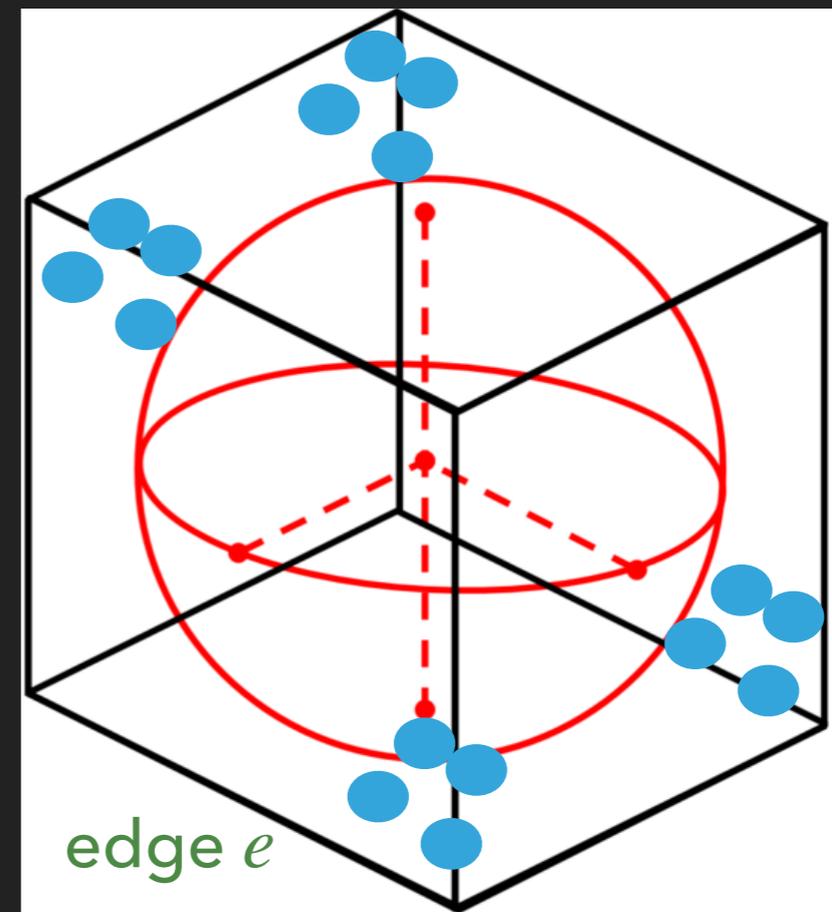
DESAFÍOS DE DATOS EN DIMENSIONES ALTAS

- ▶ Vive cerca del borde del espacio de muestra

hipercubo $C = [-e/2, e/2]^D$

hiperesfera $S = \{ ||\mathbf{x}|| > e/2 \}$

$$P(x \in S | x \in C) = \frac{\text{Vol } S}{\text{Vol } C} \sim \left(\frac{1}{2}\right)^D$$



DESAFÍOS DE DATOS EN DIMENSIONES ALTAS

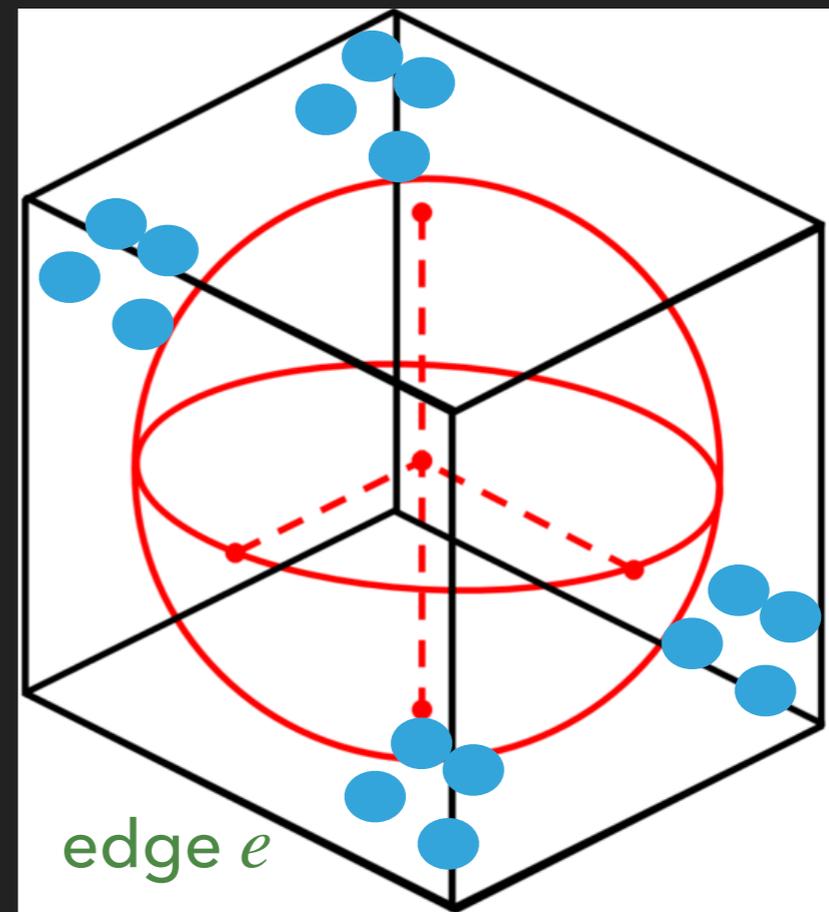
- ▶ Vive cerca del borde del espacio de muestra

hipercubo $C = [-e/2, e/2]^D$

hiperesfera $S = \{ ||\mathbf{x}|| > e/2 \}$

$$P(x \in S | x \in C) = \frac{\text{Vol } S}{\text{Vol } C} \sim \left(\frac{1}{2}\right)^D$$

$$D \rightarrow \infty \quad P \rightarrow 0$$



DESAFÍOS DE DATOS EN DIMENSIONES ALTAS (CONT...)

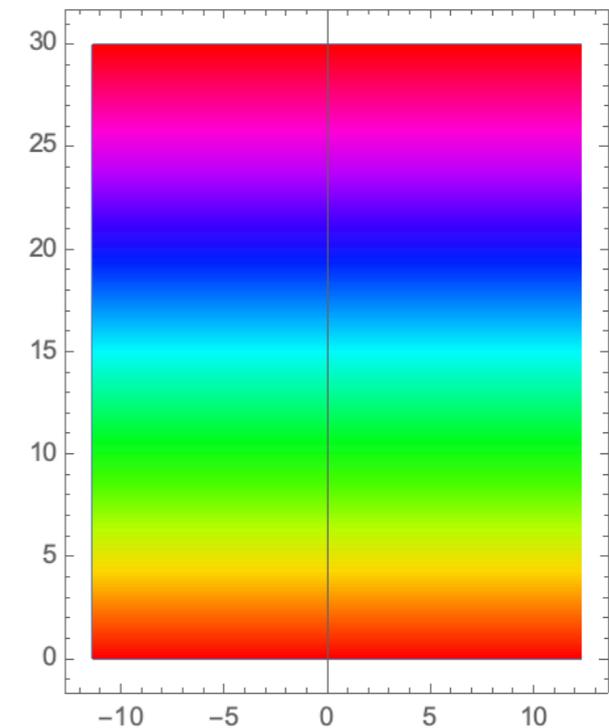
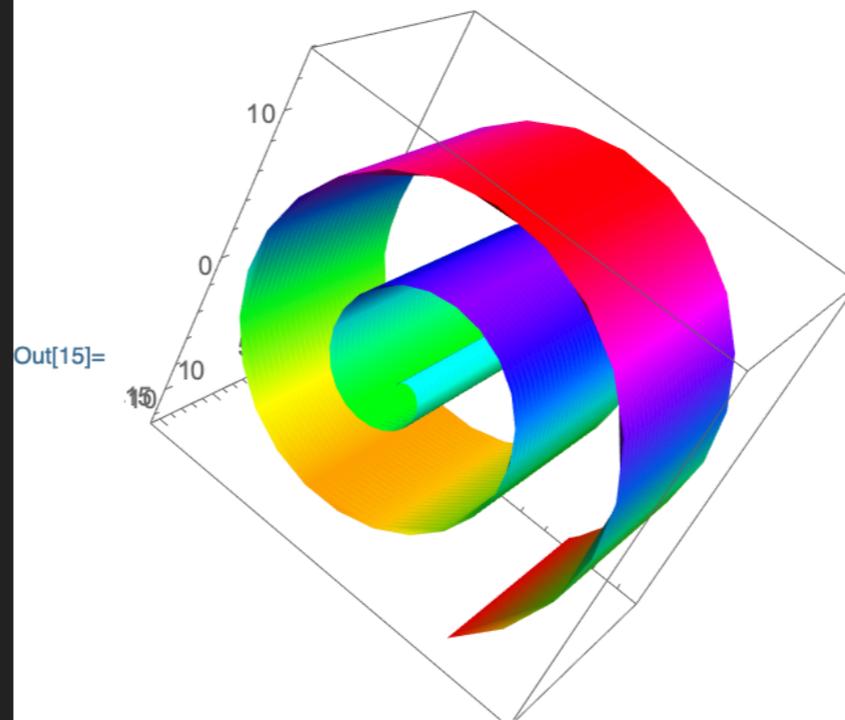
DESAFÍOS DE DATOS EN DIMENSIONES ALTAS (CONT...)

- ▶ Dimensionalidad intrínseca: número mínimo de dimensiones requeridas para captar señal de los datos

DESAFÍOS DE DATOS EN DIMENSIONES ALTAS (CONT...)

- ▶ Dimensionalidad intrínseca: número mínimo de dimensiones requeridas para captar señal de los datos

```
In[13]:= f[x1_, x2_] := {x1 Sin[x1], x1 Cos[x1], x2}
p1 = ParametricPlot3D[f[x, y], {x, 0, 15}, {y, 0, 15},
  ColorFunction -> Function[{x, y}, Hue[y]], Mesh -> None];
p2 = ParametricPlot[f[x, y][[2 ;; 3]], {x, 0, 15}, {y, 0, 30},
  ColorFunction -> Function[{x, y}, Hue[y]]];
GraphicsRow[{p1, p2}, ImageSize -> Large]
```



DESAFÍOS DE DATOS EN DIMENSIONES ALTAS (CONT...)

- ▶ Dimensionalidad intrínseca: número mínimo de dimensiones requeridas para captar señal de los datos

DESAFÍOS DE DATOS EN DIMENSIONES ALTAS (CONT...)

- ▶ Dimensionalidad intrínseca: número mínimo de dimensiones requeridas para captar señal de los datos
- ▶ Problema de amontonamiento

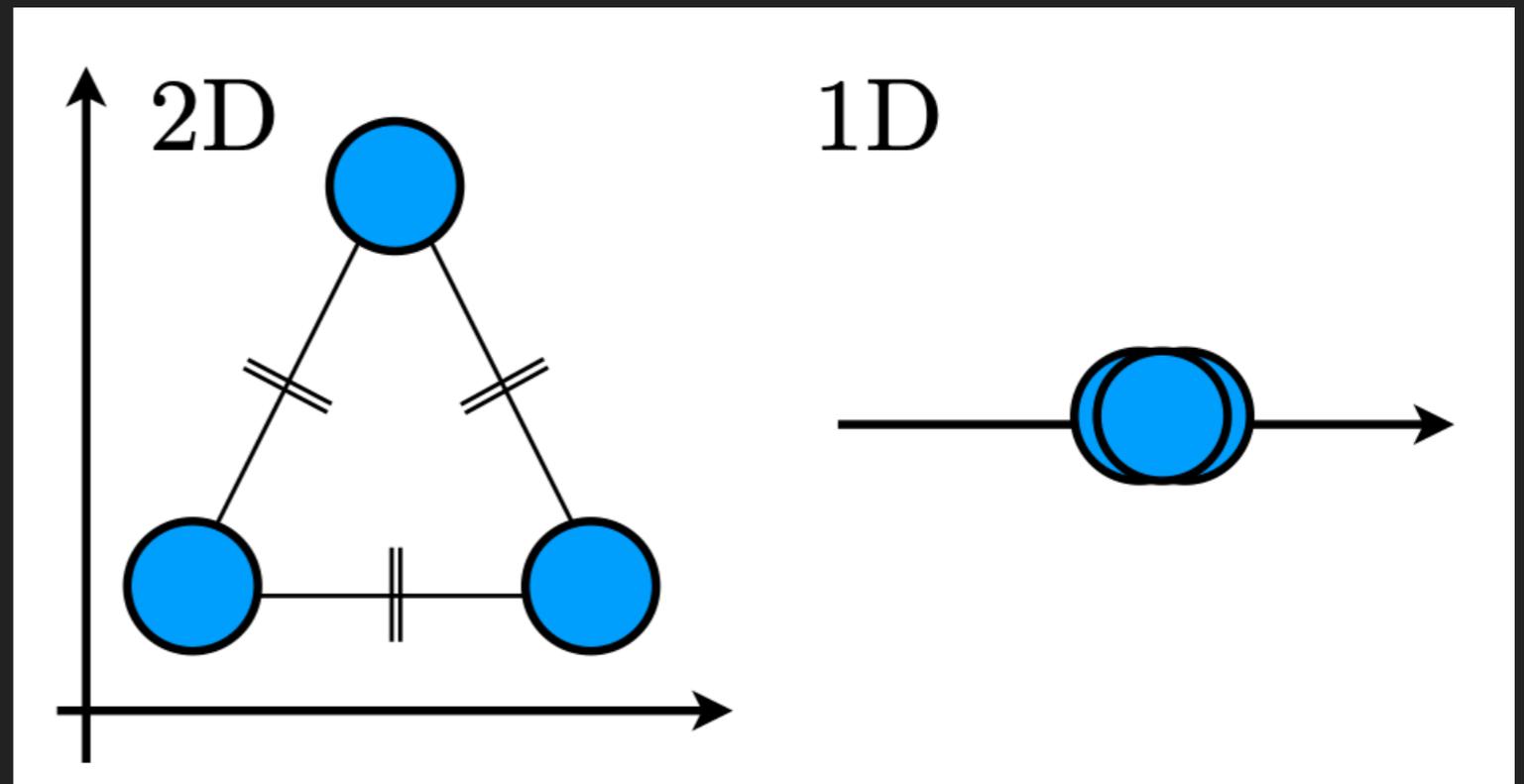
DESAFÍOS DE DATOS EN DIMENSIONES ALTAS (CONT...)

- ▶ Dimensionalidad intrínseca: número mínimo de dimensiones requeridas para captar señal de los datos
- ▶ Problema de amontonamiento

$$d_{representacion} < d_{intrinseca}$$

DESAFÍOS DE DATOS EN DIMENSIONES ALTAS (CONT...)

- ▶ Dimensionalidad intrínseca: número mínimo de dimensiones requeridas para captar señal de los datos
- ▶ Problema de amontonamiento



$$d_{representacion} < d_{intrinseca}$$

TAREAS DE APRENDIZAJE NO SUPERVISADO EN HEP

- ▶ Reducción dimensional y filtración de datos de los eventos
- Procesamiento de información
- ▶ Uso de modelos generativos para hacer simulaciones eficientes y rápidas
- ▶ Uso de modelos de aprendizaje no supervisado, para mejorar aquellos modelos de aprendizaje supervisado

APRENDIZAJE POR REFUERZO = REINFORCEMENT LEARNING

nature

Article | Published: 27 January 2016

Mastering the game of Go with deep neural networks and tree search

David Silver [✉](#), Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, [George van den Driessche](#), Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel & Demis Hassabis [✉](#)

Nature 529, 484–489(2016) | [Cite this article](#)

84k Accesses | 2525 Citations | 3133 Altmetric | [Metrics](#)

Abstract

The game of Go has long been viewed as the most challenging of classic games for artificial intelligence owing to its enormous search space and the difficulty of evaluating board positions and moves. Here we introduce a new approach to computer Go that uses ‘value networks’ to evaluate board positions and ‘policy networks’ to select moves. These deep neural networks are trained by a novel combination of supervised learning from human expert games, and reinforcement learning from games of self-play. Without any lookahead search, the neural networks play Go at the level of state-of-the-art Monte Carlo tree search programs that simulate thousands of random games of self-play. We also introduce a new search algorithm that combines Monte Carlo simulation with value and policy networks. Using this search algorithm, our program AlphaGo achieved a 99.8% winning rate against other Go programs, and defeated the human European Go champion by 5 games to 0. This is the first time that a computer program has defeated a human professional player in the full-sized game of Go, a feat previously thought to be at least a decade away.

Google AI defeats human Go champion

🕒 25 May 2017



[Share](#)



REUTERS

Chinese Go player Ke Jie has lost two games to AlphaGo

Google's DeepMind AlphaGo artificial intelligence has defeated the world's number one Go player Ke Jie.

APRENDIZAJE POR REFUERZO

TAREAS DE APRENDIZAJE POR REFUERZO EN FÍSICA

APRENDIZAJE POR REFUERZO

TAREAS DE APRENDIZAJE POR REFUERZO EN FÍSICA

TAREAS DE APRENDIZAJE POR REFUERZO EN FÍSICA

- ▶ **Computación cuántica**

TAREAS DE APRENDIZAJE POR REFUERZO EN FÍSICA

- ▶ **Computación cuántica**
 - ▶ Control y preparación de Qubits

TAREAS DE APRENDIZAJE POR REFUERZO EN FÍSICA

▶ Computación cuántica

▶ Control y preparación de Qubits

Reinforcement Learning in Different Phases of Quantum Control

Marin Bukov,^{1,*} Alexandre G. R. Day,^{1,†} Dries Sels,^{1,2} Phillip Weinberg,¹ Anatoli Polkovnikov,¹ and Pankaj Mehta¹

¹*Department of Physics, Boston University,*

590 Commonwealth Avenue, Boston, Massachusetts 02215, USA 

²*Theory of quantum and complex systems, Universiteit Antwerpen, B-2610 Antwerpen, Belgium*

 (Received 12 January 2018; revised manuscript received 1 August 2018; published 27 September 2018)

The ability to prepare a physical system in a desired quantum state is central to many areas of physics such as nuclear magnetic resonance, cold atoms, and quantum computing. Yet, preparing states quickly and with high fidelity remains a formidable challenge. In this work, we implement cutting-edge reinforcement learning (RL) techniques and show that their performance is comparable to optimal control methods in the task of finding short, high-fidelity driving protocol from an initial to a target state in nonintegrable many-body quantum systems of interacting qubits. RL methods learn about the underlying physical system solely through a single scalar reward (the fidelity of the resulting state) calculated from numerical simulations of the physical system. We further show that quantum-state manipulation viewed as an optimization problem exhibits a spin-glass-like phase transition in the space of protocols as a function of the protocol duration. Our RL-aided approach helps identify variational protocols with nearly optimal fidelity, even in the glassy phase, where optimal state manipulation is exponentially hard. This study highlights the potential usefulness of RL for applications in out-of-equilibrium quantum physics.

DOI: [10.1103/PhysRevX.8.031086](https://doi.org/10.1103/PhysRevX.8.031086)

Subject Areas: Condensed Matter Physics,
Quantum Physics, Statistical Physics

TAREAS DE APRENDIZAJE POR REFUERZO EN FÍSICA

▶ Computación cuántica

▶ Control y preparación de Qubits

▶ Corrección de errores cuánticos

Reinforcement Learning in Different Phases of Quantum Control

Marin Bukov,^{1,*} Alexandre G. R. Day,^{1,†} Dries Sels,^{1,2} Phillip Weinberg,¹ Anatoli Polkovnikov,¹ and Pankaj Mehta¹

¹*Department of Physics, Boston University,*

590 Commonwealth Avenue, Boston, Massachusetts 02215, USA 

²*Theory of quantum and complex systems, Universiteit Antwerpen, B-2610 Antwerpen, Belgium*

 (Received 12 January 2018; revised manuscript received 1 August 2018; published 27 September 2018)

The ability to prepare a physical system in a desired quantum state is central to many areas of physics such as nuclear magnetic resonance, cold atoms, and quantum computing. Yet, preparing states quickly and with high fidelity remains a formidable challenge. In this work, we implement cutting-edge reinforcement learning (RL) techniques and show that their performance is comparable to optimal control methods in the task of finding short, high-fidelity driving protocol from an initial to a target state in nonintegrable many-body quantum systems of interacting qubits. RL methods learn about the underlying physical system solely through a single scalar reward (the fidelity of the resulting state) calculated from numerical simulations of the physical system. We further show that quantum-state manipulation viewed as an optimization problem exhibits a spin-glass-like phase transition in the space of protocols as a function of the protocol duration. Our RL-aided approach helps identify variational protocols with nearly optimal fidelity, even in the glassy phase, where optimal state manipulation is exponentially hard. This study highlights the potential usefulness of RL for applications in out-of-equilibrium quantum physics.

DOI: [10.1103/PhysRevX.8.031086](https://doi.org/10.1103/PhysRevX.8.031086)

Subject Areas: Condensed Matter Physics,
Quantum Physics, Statistical Physics

TAREAS DE APRENDIZAJE POR REFUERZO EN FÍSICA

▶ Computación cuántica

▶ Control y preparación de Qubits

▶ Corrección de errores cuánticos

Reinforcement Learning in Different Phases of Quantum Control

Marin Bukov,^{1,*} Alexandre G. R. Day,^{1,†} Dries Sels,^{1,2} Phillip Weinberg,¹ Anatoli Polkovnikov,¹ and Pankaj Mehta¹

¹*Department of Physics, Boston University,*

590 Commonwealth Avenue, Boston, Massachusetts 02215, USA

²*Theory of quantum and complex systems, Universiteit Antwerpen, B-2610 Antwerpen, Belgium*

Ⓞ (Received 12 January 2018; revised manuscript received 1 August 2018; published 27 September 2018)

The ability to prepare a physical system in a desired quantum state is central to many areas of physics such as nuclear magnetic resonance, cold atoms, and quantum computing. Yet, preparing states quickly and with high fidelity remains a formidable challenge. In this work, we implement cutting-edge reinforcement learning (RL) techniques and show that their performance is comparable to optimal control methods in the task of finding short, high-fidelity driving protocol from an initial to a target state in nonintegrable many-body quantum systems of interacting qubits. RL methods learn about the underlying physical system solely through a single scalar reward (the fidelity of the resulting state) calculated from numerical simulations of the physical system. We further show that quantum-state manipulation viewed as an optimization problem exhibits a spin-glass-like phase transition in the space of protocols as a function of the protocol duration. Our RL-aided approach helps identify variational protocols with nearly optimal fidelity, even in the glassy phase, where optimal state manipulation is exponentially hard. This study highlights the potential usefulness of RL for applications in out-of-equilibrium quantum physics.

DOI: [10.1103/PhysRevX.8.031086](https://doi.org/10.1103/PhysRevX.8.031086)

Subject Areas: Condensed Matter Physics,
Quantum Physics, Statistical Physics

Reinforcement Learning Decoders for Fault-Tolerant Quantum Computation

Ryan Sweke,¹ Markus S. Kesselring,¹ Evert P. L. van Nieuwenburg,² and Jens Eisert^{1,3}

¹*Dahlem Center for Complex Quantum Systems, Freie Universität Berlin, 14195 Berlin, Germany*

²*Institute for Quantum Information and Matter, Caltech, Pasadena, CA 91125, USA*

³*Department of Mathematics and Computer Science, Freie Universität Berlin, 14195 Berlin*

(Dated: October 18, 2018)

Topological error correcting codes, and particularly the surface code, currently provide the most feasible roadmap towards large-scale fault-tolerant quantum computation. As such, obtaining fast and flexible decoding algorithms for these codes, within the experimentally relevant context of faulty syndrome measurements, is of critical importance. In this work, we show that the problem of decoding such codes, in the full fault-tolerant setting, can be naturally reformulated as a process of repeated interactions between a decoding agent and a code environment, to which the machinery of reinforcement learning can be applied to obtain decoding agents. As a demonstration, by using deepQ learning, we obtain fast decoding agents for the surface code, for a variety of noise-models.

IMPLEMENTACIONES

IMPLEMENTACIONES

IMPLEMENTACIONES

SIMULACIONES EN HEP

SIMULACIONES EN HEP

- ▶ Generadores de evento por Monte Carlo

SIMULACIONES EN HEP

- ▶ Generadores de evento por Monte Carlo
- ▶ Ariadne, Herwig++, Pythia 8 and Sherpa, Delphes

SIMULACIONES EN HEP

- ▶ Generadores de evento por Monte Carlo
- ▶ Ariadne, Herwig++, Pythia 8 and Sherpa, Delphes
 - ▶ Expansion perturbativa de diagramas de Feynmann

SIMULACIONES EN HEP

- ▶ Generadores de evento por Monte Carlo
- ▶ Ariadne, Herwig++, Pythia 8 and Sherpa, Delphes
 - ▶ Expansion perturbativa de diagramas de Feynmann
 - ▶ Modelos fenomenologicos de radiación

SIMULACIONES EN HEP

- ▶ Generadores de evento por Monte Carlo
- ▶ Ariadne, Herwig++, Pythia 8 and Sherpa, Delphes
 - ▶ Expansion perturbativa de diagramas de Feynmann
 - ▶ Modelos fenomenologicos de radiación
 - ▶ Modelos de interacción de partículas con materiales de los detectores

MODELOS

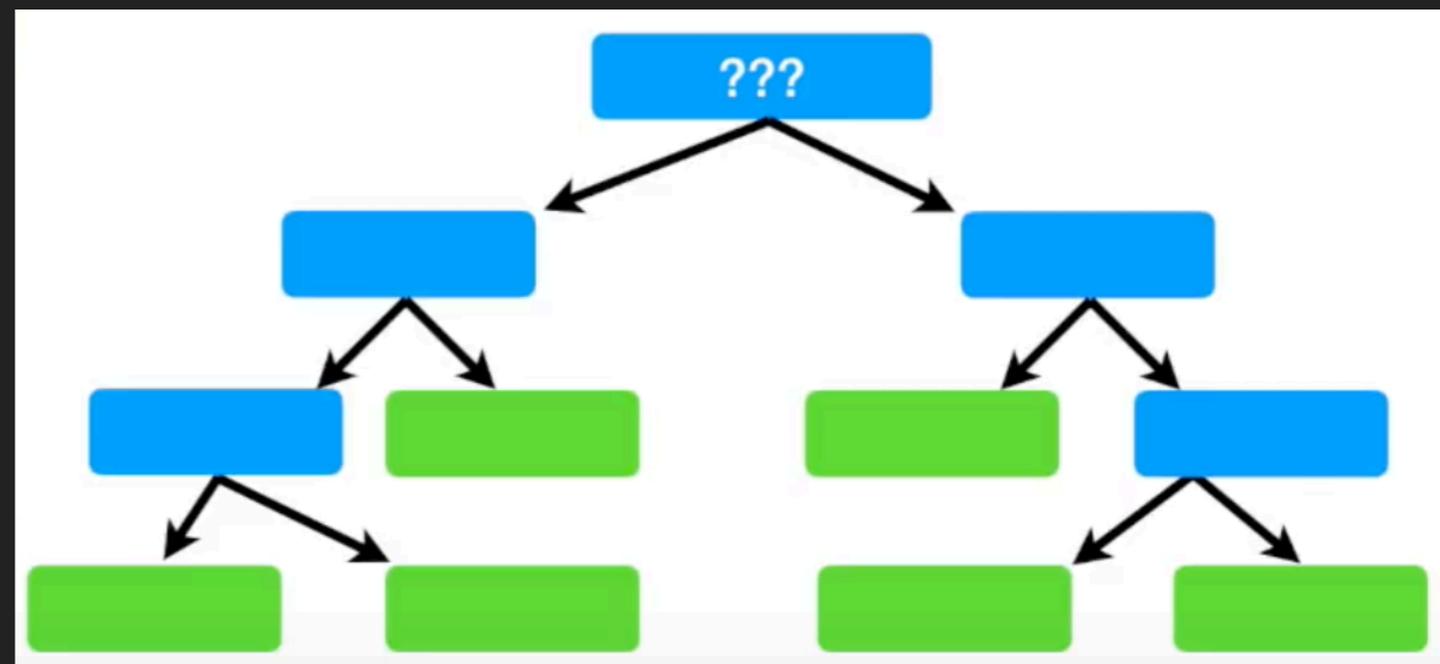
ARBOLES DE DECISION

ARBOLES DE DECISION

Chest Pain	Good Blood Circulation	Blocked Arteries	Heart Disease
No	No	No	No
Yes	Yes	Yes	Yes
Yes	Yes	No	No
Yes	No	???	Yes
etc...	etc...	etc...	etc...

ARBOLES DE DECISION

Chest Pain	Good Blood Circulation	Blocked Arteries	Heart Disease
No	No	No	No
Yes	Yes	Yes	Yes
Yes	Yes	No	No
Yes	No	???	Yes
etc...	etc...	etc...	etc...

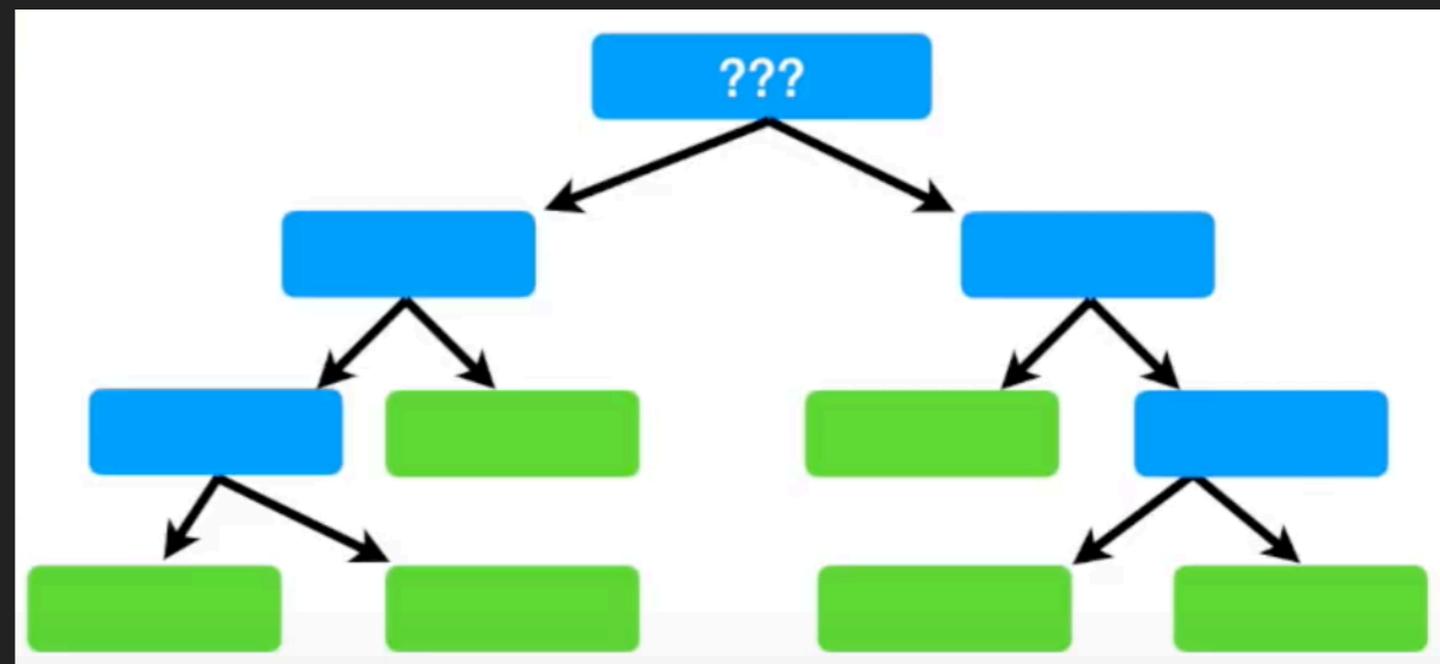


ARBOLES DE DECISION

- ▶ Impureza de Gini usado para definir nodos del árbol y valores de corte

$$g = 1 - \sum_i^n f_i^2 ; i = 1, \dots, m$$

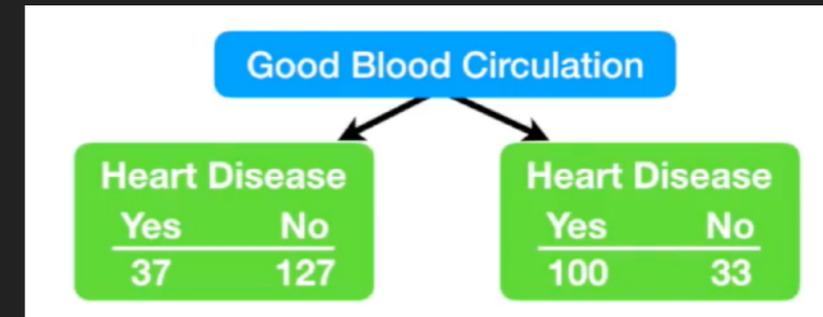
Chest Pain	Good Blood Circulation	Blocked Arteries	Heart Disease
No	No	No	No
Yes	Yes	Yes	Yes
Yes	Yes	No	No
Yes	No	???	Yes
etc...	etc...	etc...	etc...



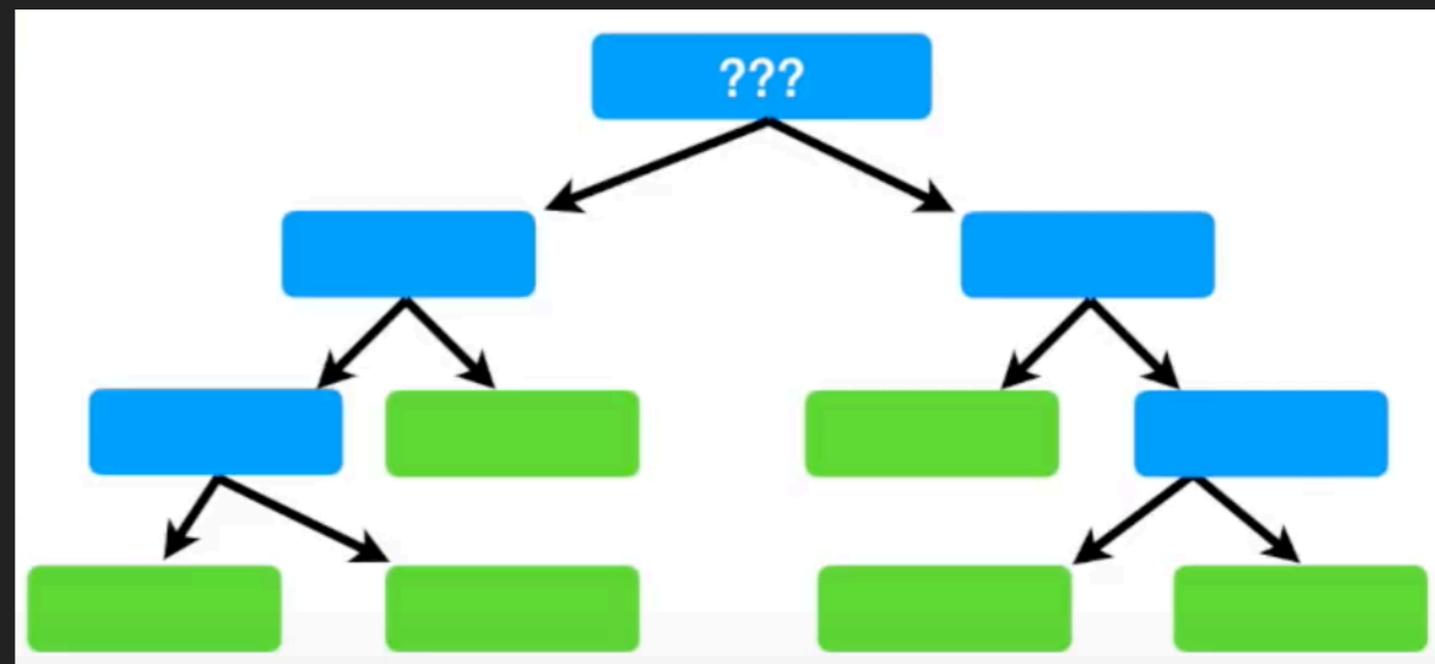
ARBOLES DE DECISION

- ▶ Impureza de Gini usado para definir nodos del árbol y valores de corte

$$g = 1 - \sum_i^n f_i^2 ; i = 1, \dots, m$$



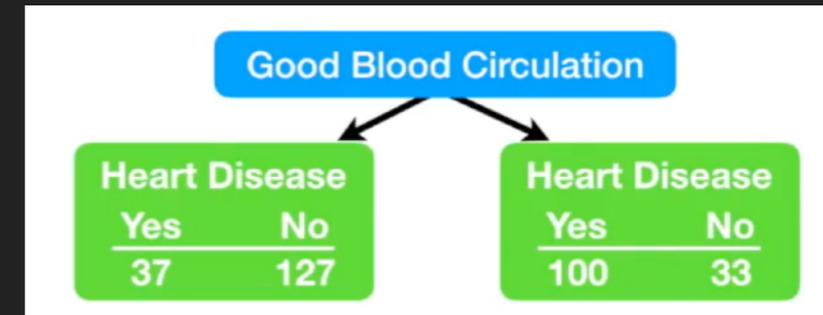
Chest Pain	Good Blood Circulation	Blocked Arteries	Heart Disease
No	No	No	No
Yes	Yes	Yes	Yes
Yes	Yes	No	No
Yes	No	???	Yes
etc...	etc...	etc...	etc...



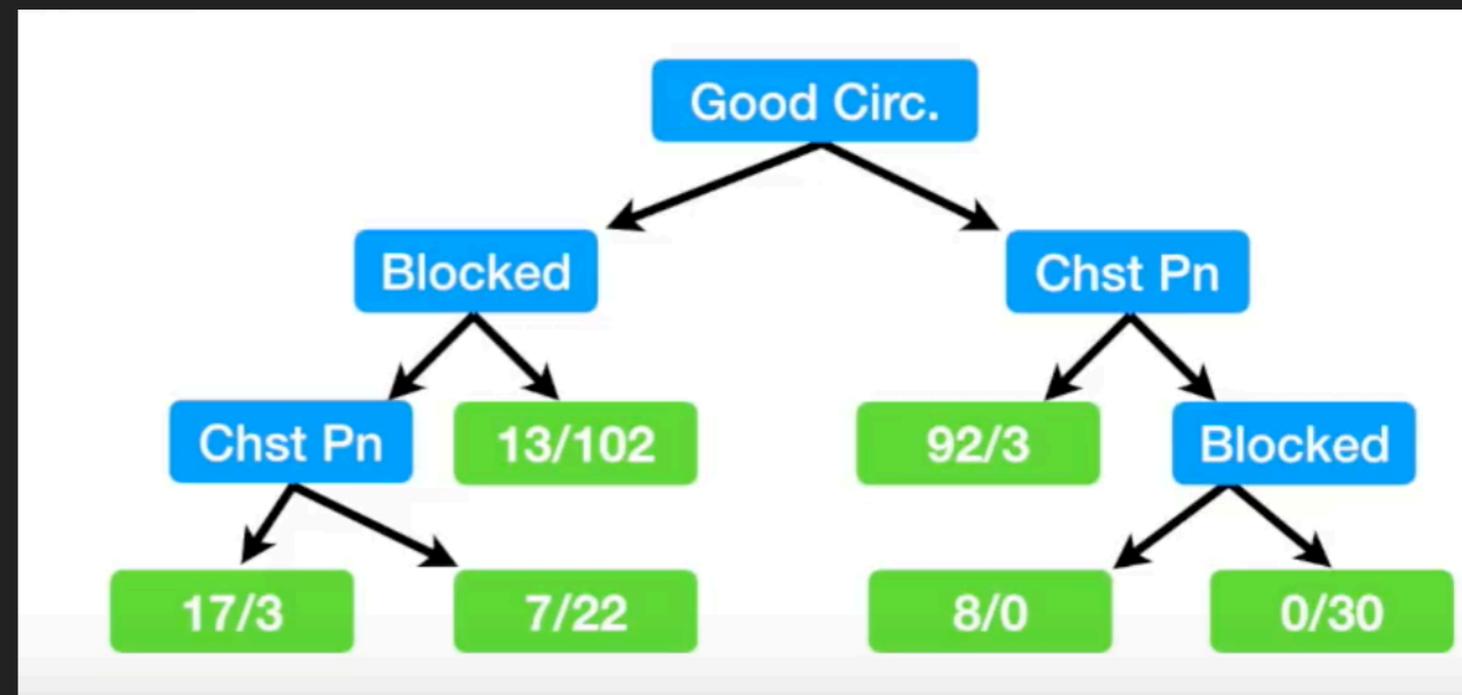
ARBOLES DE DECISION

- ▶ Impureza de Gini usado para definir nodos del árbol y valores de corte

$$g = 1 - \sum_i^n f_i^2 ; i = 1, \dots, m$$



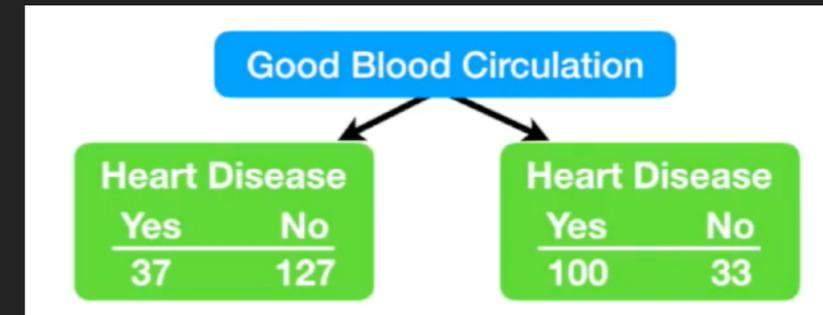
Chest Pain	Good Blood Circulation	Blocked Arteries	Heart Disease
No	No	No	No
Yes	Yes	Yes	Yes
Yes	Yes	No	No
Yes	No	???	Yes
etc...	etc...	etc...	etc...



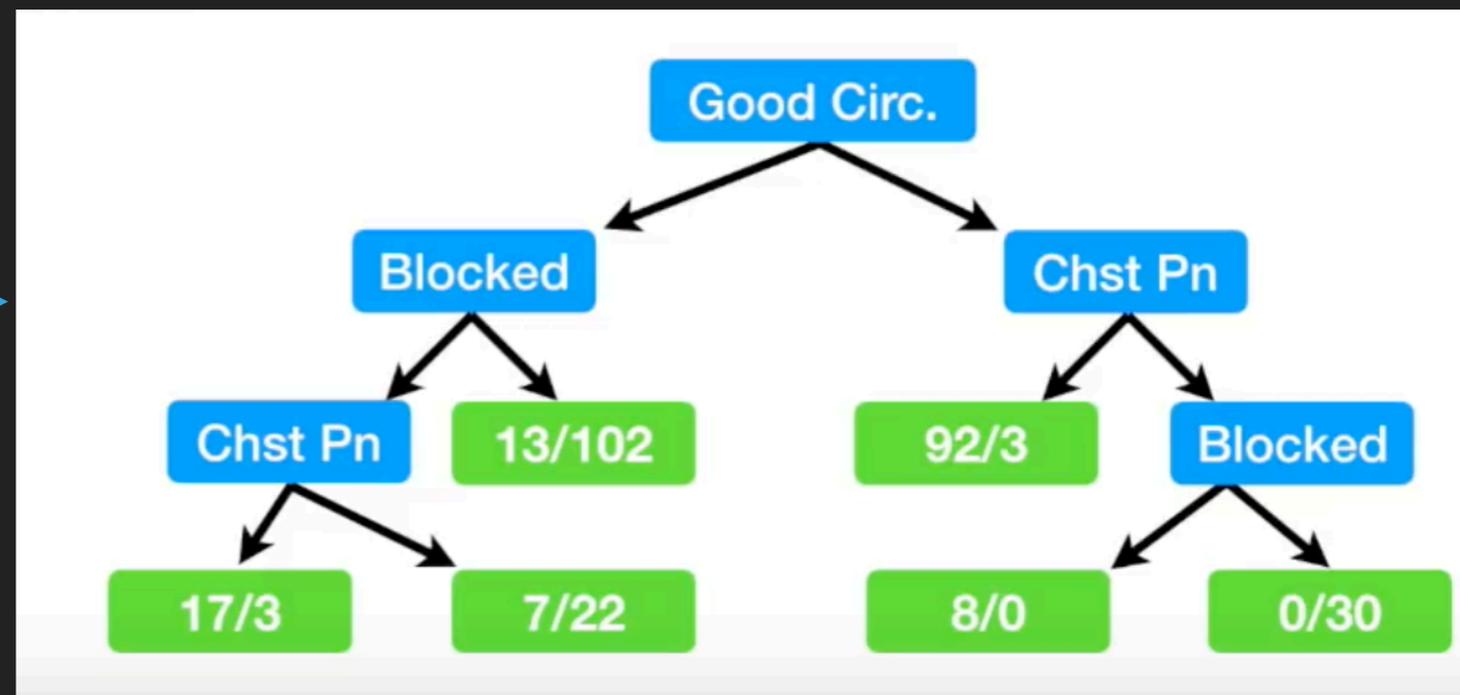
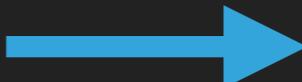
ARBOLES DE DECISION

- ▶ Impureza de Gini usado para definir nodos del árbol y valores de corte

$$g = 1 - \sum_i^n f_i^2 ; i = 1, \dots, m$$



Chest Pain	Good Blood Circulation	Blocked Arteries	Heart Disease
No	No	No	No
Yes	Yes	Yes	Yes
Yes	Yes	No	No
Yes	No	???	Yes
etc...	etc...	etc...	etc...



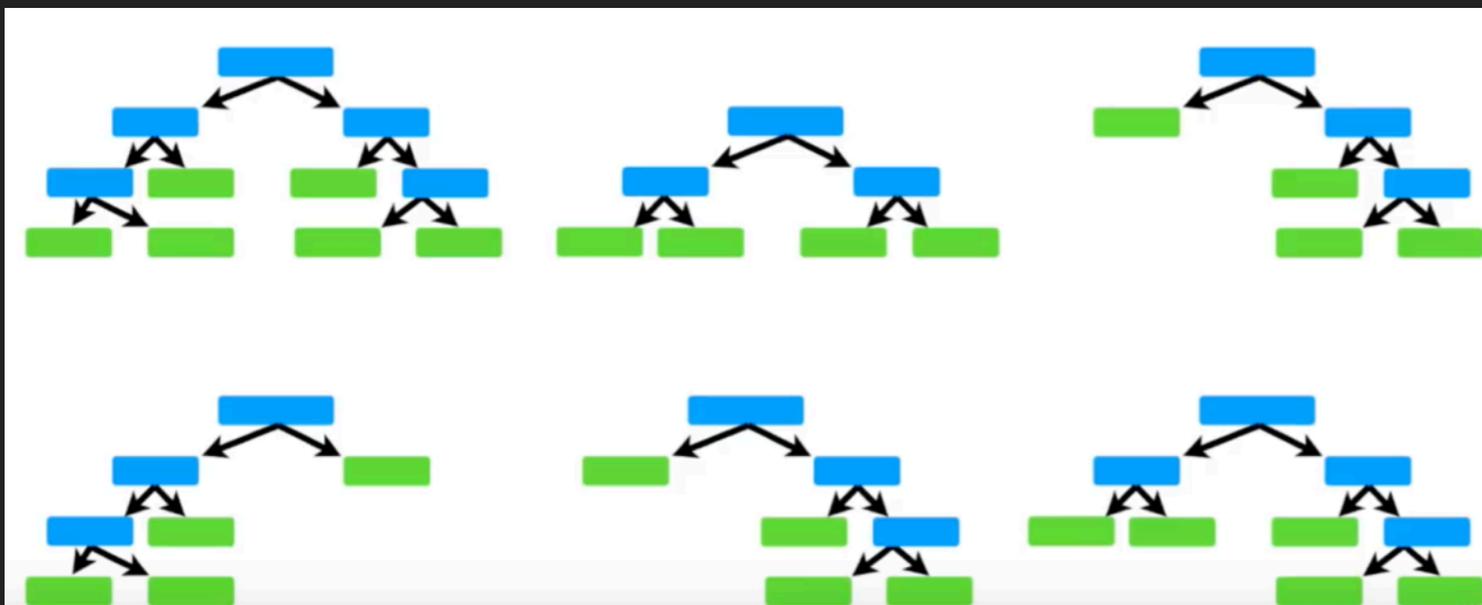
RANDOM FORESTS

RANDOM FORESTS

- ▶ Crear arboles de decisión con subconjuntos aleatorios de los datos, y subconjuntos aleatorios de variables

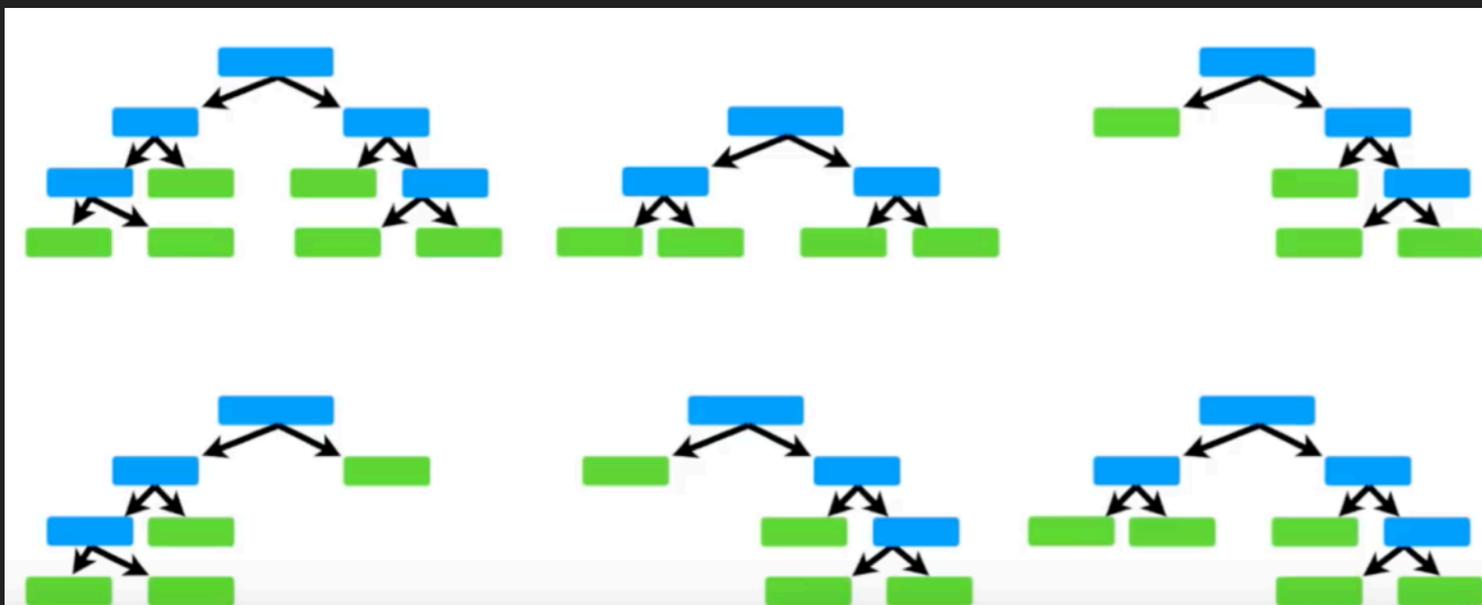
RANDOM FORESTS

- ▶ Crear arboles de decisión con subconjuntos aleatorios de los datos, y subconjuntos aleatorios de variables



RANDOM FORESTS

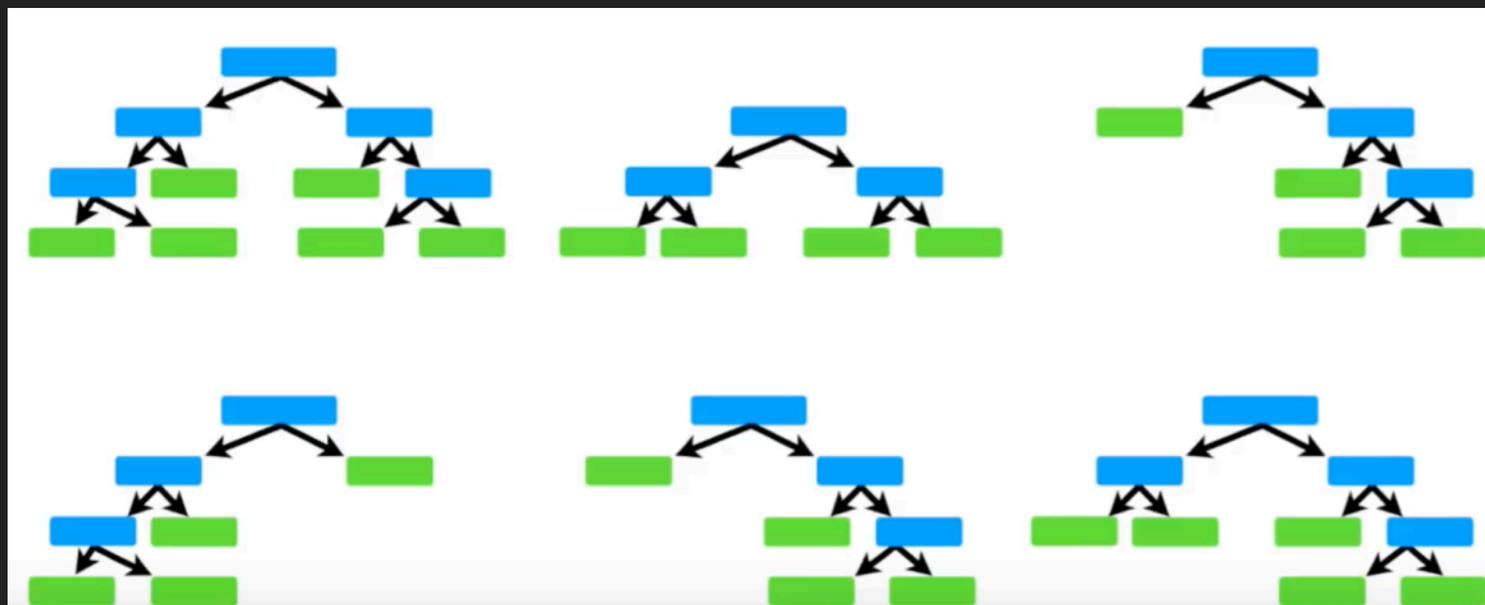
- ▶ Crear arboles de decisión con subconjuntos aleatorios de los datos, y subconjuntos aleatorios de variables



Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
Yes	No	No	168	YES

RANDOM FORESTS

- ▶ Crear arboles de decisión con subconjuntos aleatorios de los datos, y subconjuntos aleatorios de variables

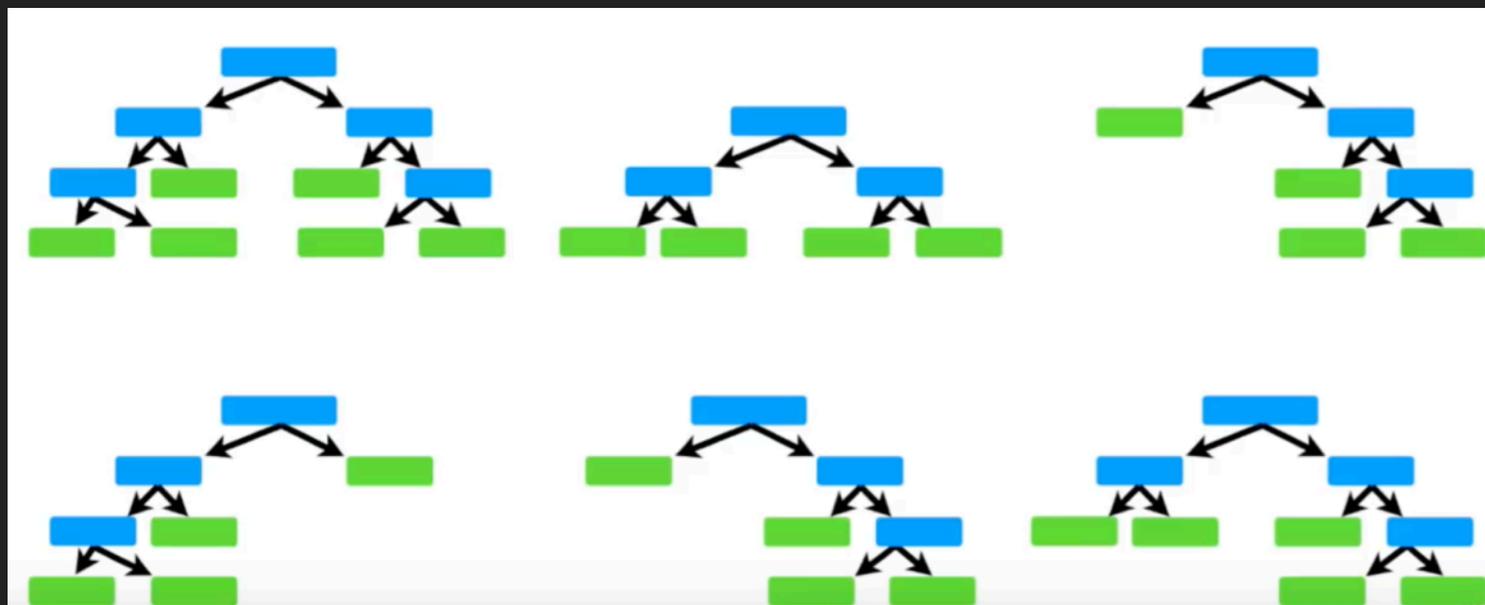


Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
Yes	No	No	168	YES

Nuevo ejemplo o muestra

RANDOM FORESTS

- ▶ Crear arboles de decisión con subconjuntos aleatorios de los datos, y subconjuntos aleatorios de variables



https://www.youtube.com/playlist?list=PLbh5JKOoLUICTaGLRoHQDuF_7q2GfuJF

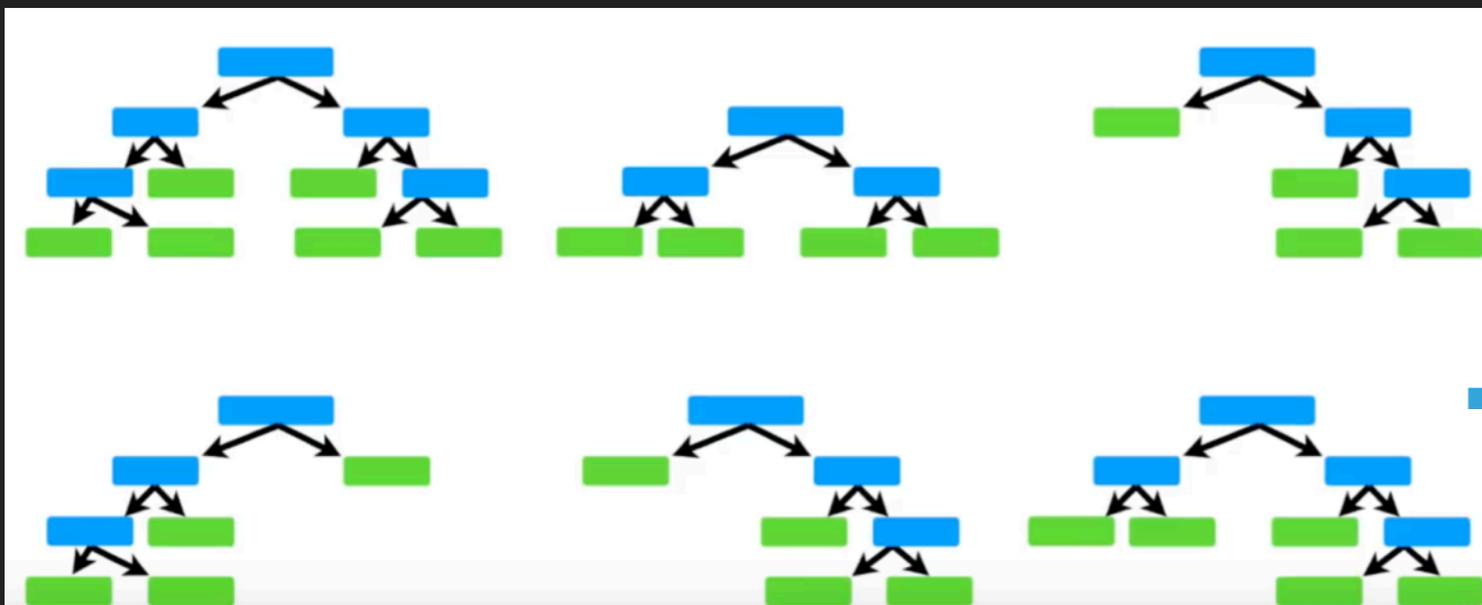
Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
Yes	No	No	168	YES

Nuevo ejemplo o muestra

Heart Disease	
Yes	No
5	1

RANDOM FORESTS

- ▶ Crear arboles de decisión con subconjuntos aleatorios de los datos, y subconjuntos aleatorios de variables



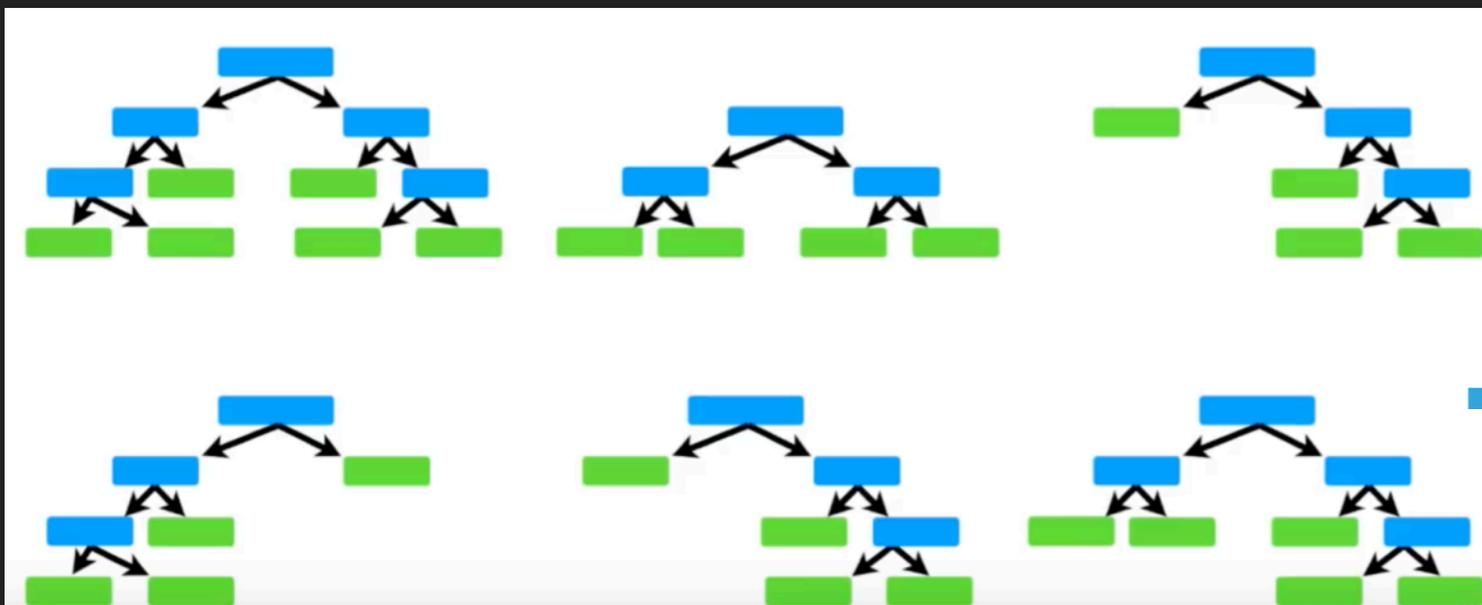
Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
Yes	No	No	168	YES

Nuevo ejemplo o muestra



RANDOM FORESTS

- ▶ Crear arboles de decisión con subconjuntos aleatorios de los datos, y subconjuntos aleatorios de variables
- ▶ Bagging

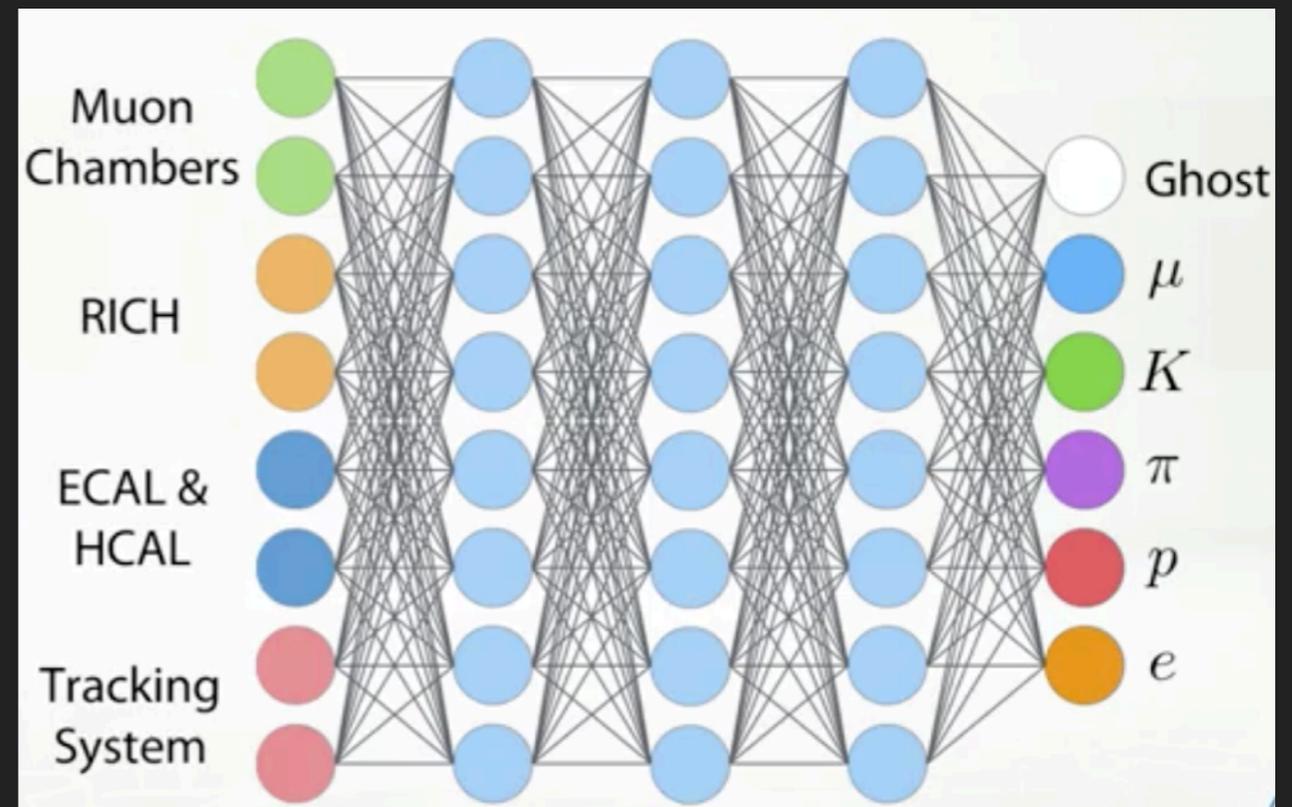


Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
Yes	No	No	168	YES

Nuevo ejemplo o muestra

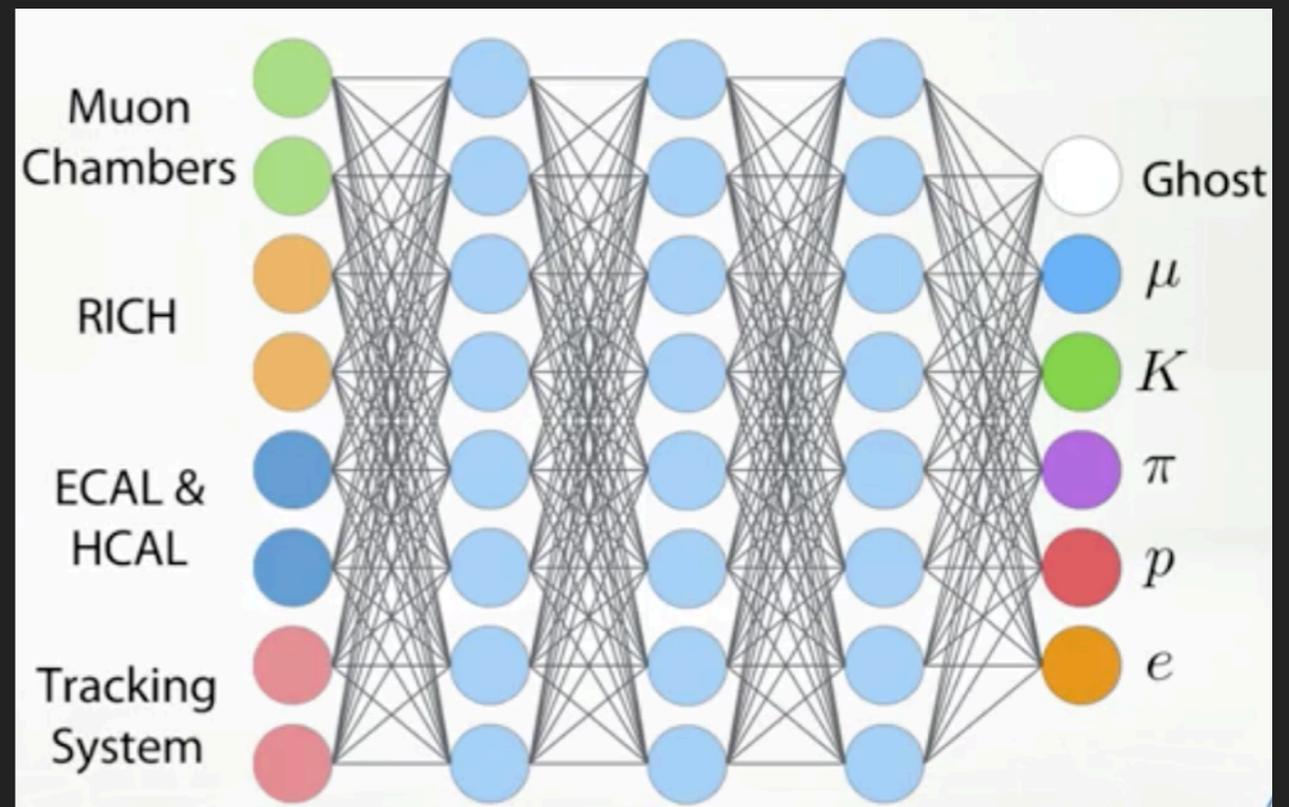


APRENDIZAJE AUTOMATIZADO EN IDENTIFICACIÓN DE PARTÍCULAS



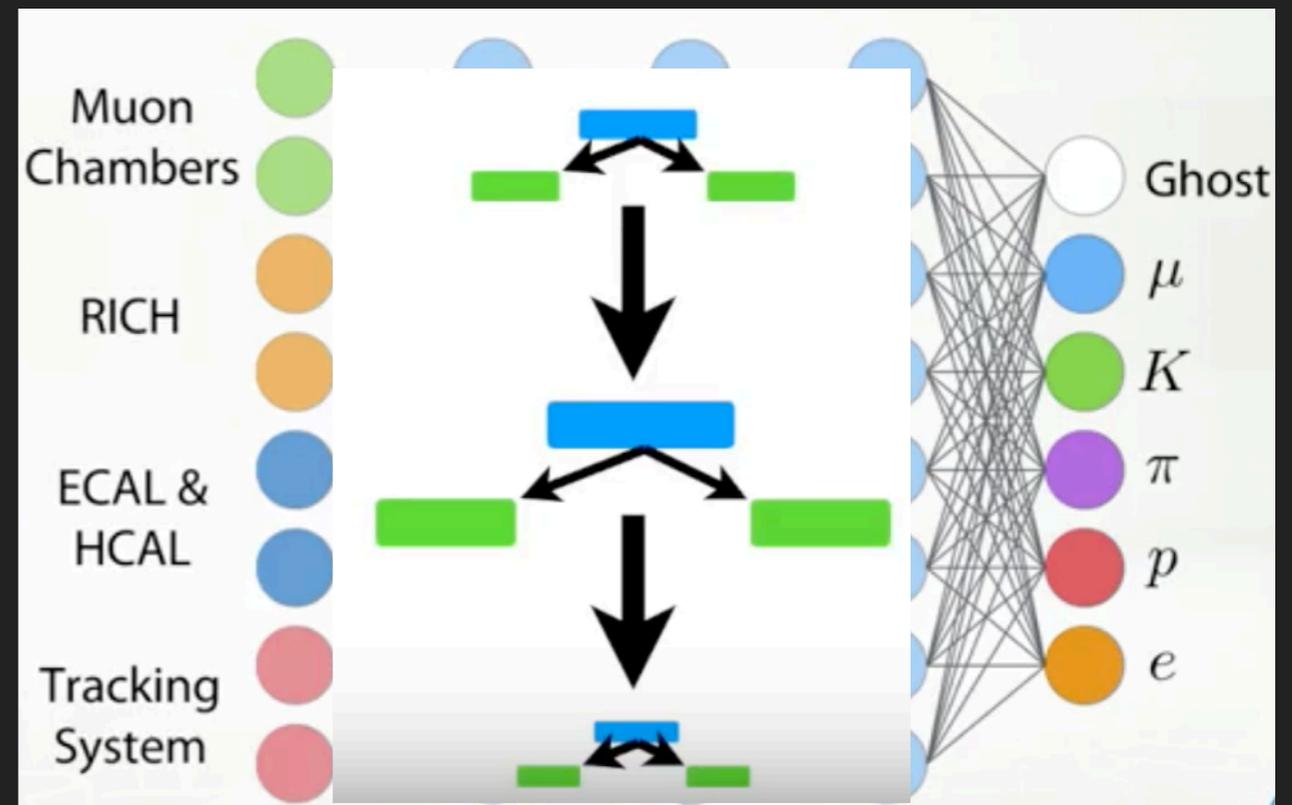
APRENDIZAJE AUTOMATIZADO EN IDENTIFICACIÓN DE PARTÍCULAS

- ▶ Identificación de partículas como un problema de clasificación de múltiples clases
- ▶ Algoritmos muy usados en las últimas décadas los BDT (Árboles de Decisión Potenciados por Gradiente) en las décadas de 1990s y 2000s (Carleo et. al. 2019)



APRENDIZAJE AUTOMATIZADO EN IDENTIFICACIÓN DE PARTÍCULAS

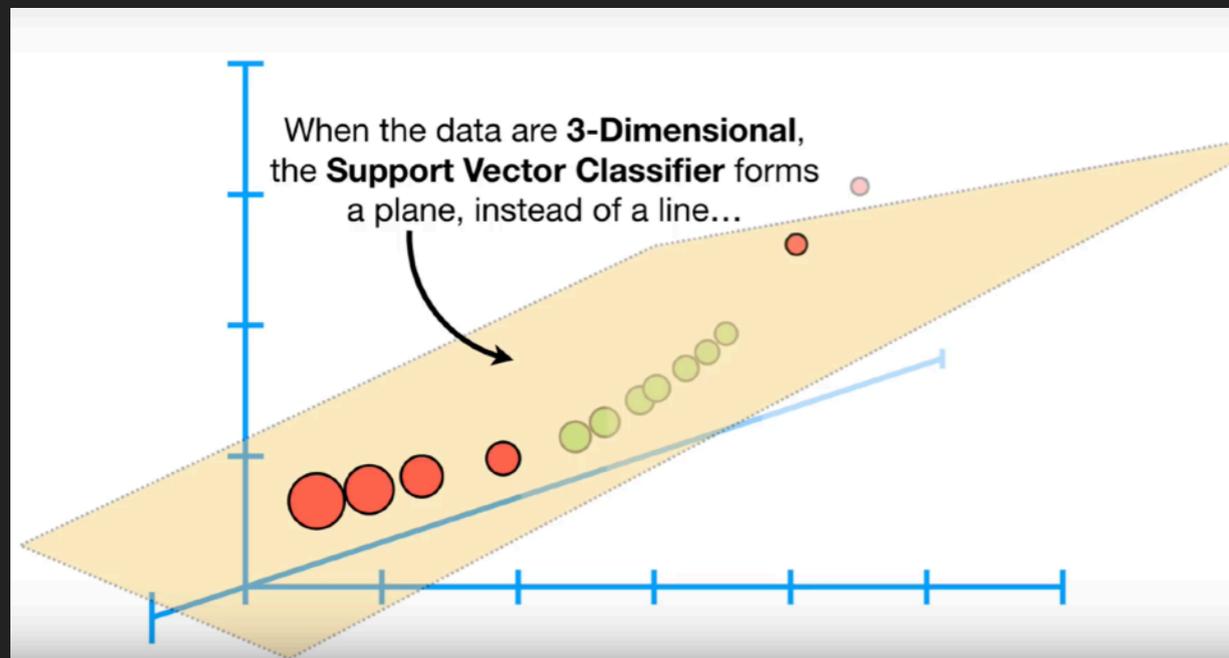
- ▶ Identificación de partículas como un problema de clasificación de múltiples clases
- ▶ Algoritmos muy usados en las últimas décadas los BDT (Árboles de Decisión Potenciados por Gradiente) en las décadas de 1990s y 2000s (Carleo et. al. 2019)



SEÑAL VS BACKGROUND

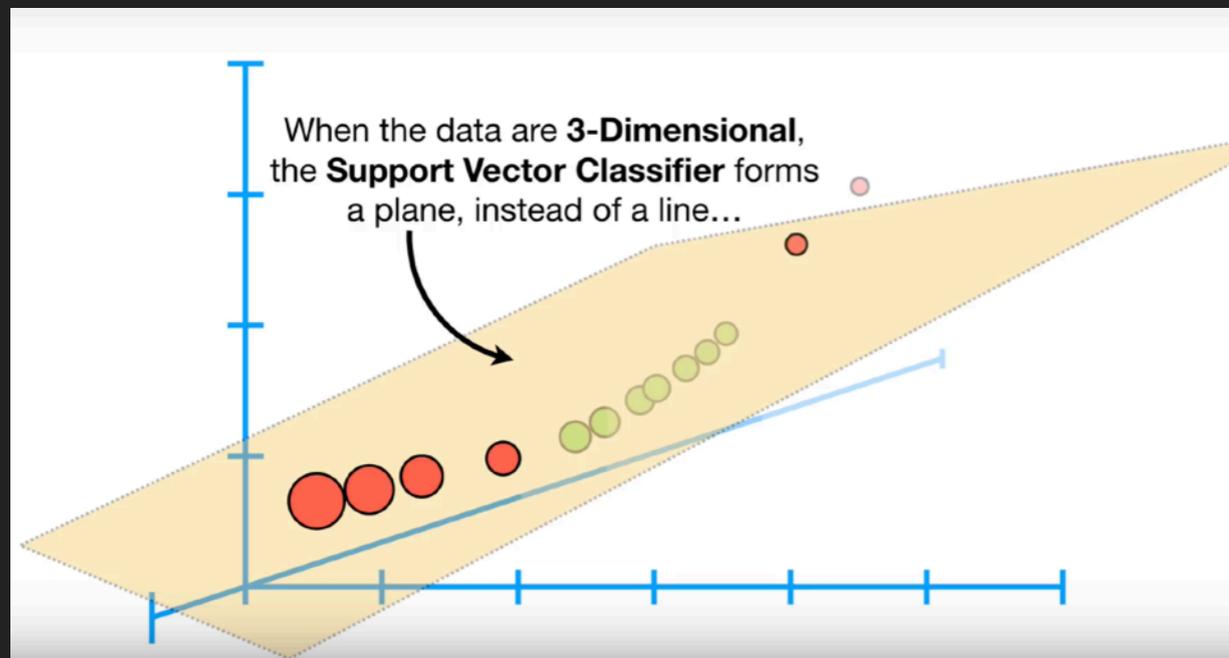
- ▶ Reconocimiento de una señal no vista anteriormente sobre un gran volumen de datos ya estudiados (Background)
- ▶ Triggering - sistemas de reducción de datos al analizar señales de interés sobre un Background
- ▶ Etiquetamiento de jets / Jet tagging

SUPPORT VECTOR MACHINES



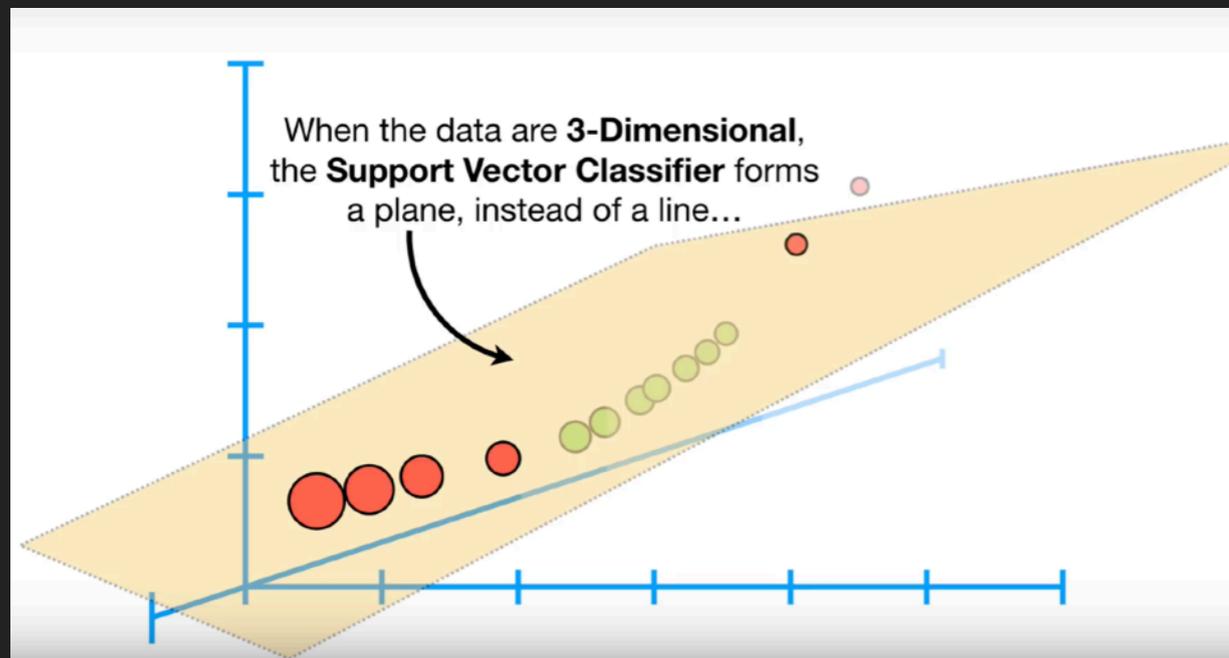
SUPPORT VECTOR MACHINES

- ▶ Separación de datos por hiper-planos



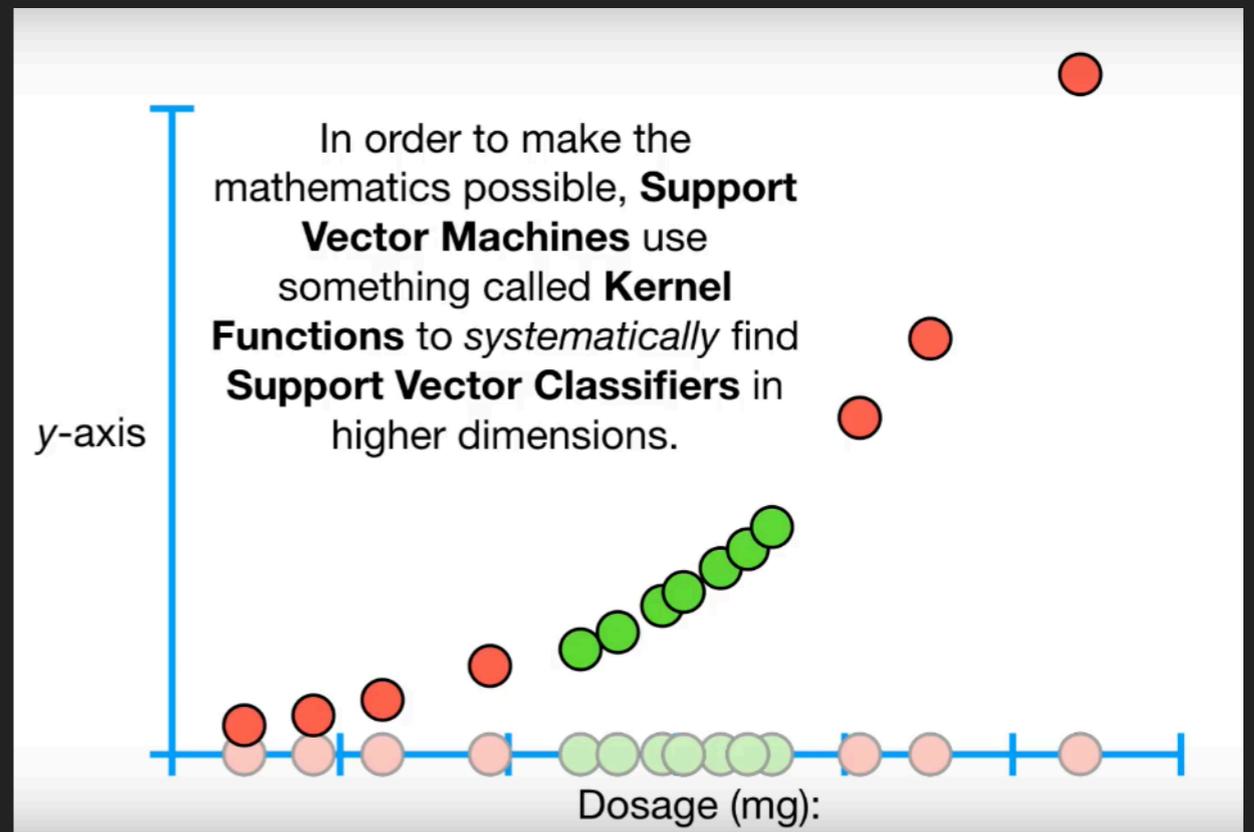
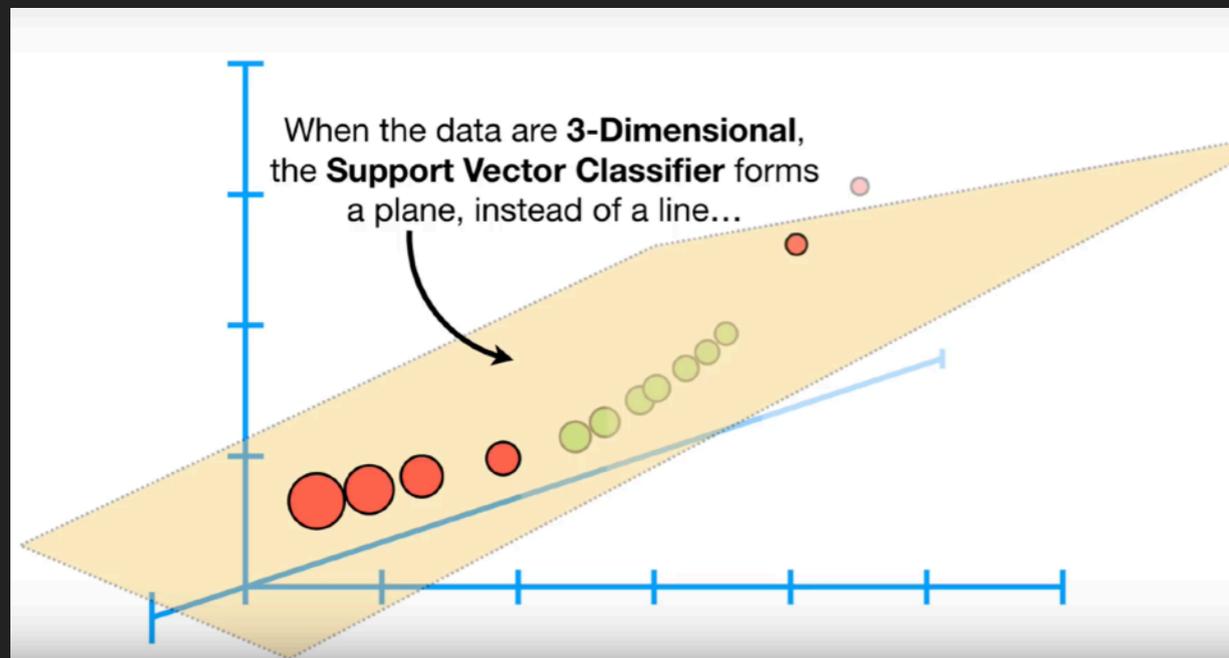
SUPPORT VECTOR MACHINES

- ▶ Separación de datos por hiper-planos
- ▶ Usar Funciones de Kernel para encontrar un Support Vector Classifier, por validación cruzada



SUPPORT VECTOR MACHINES

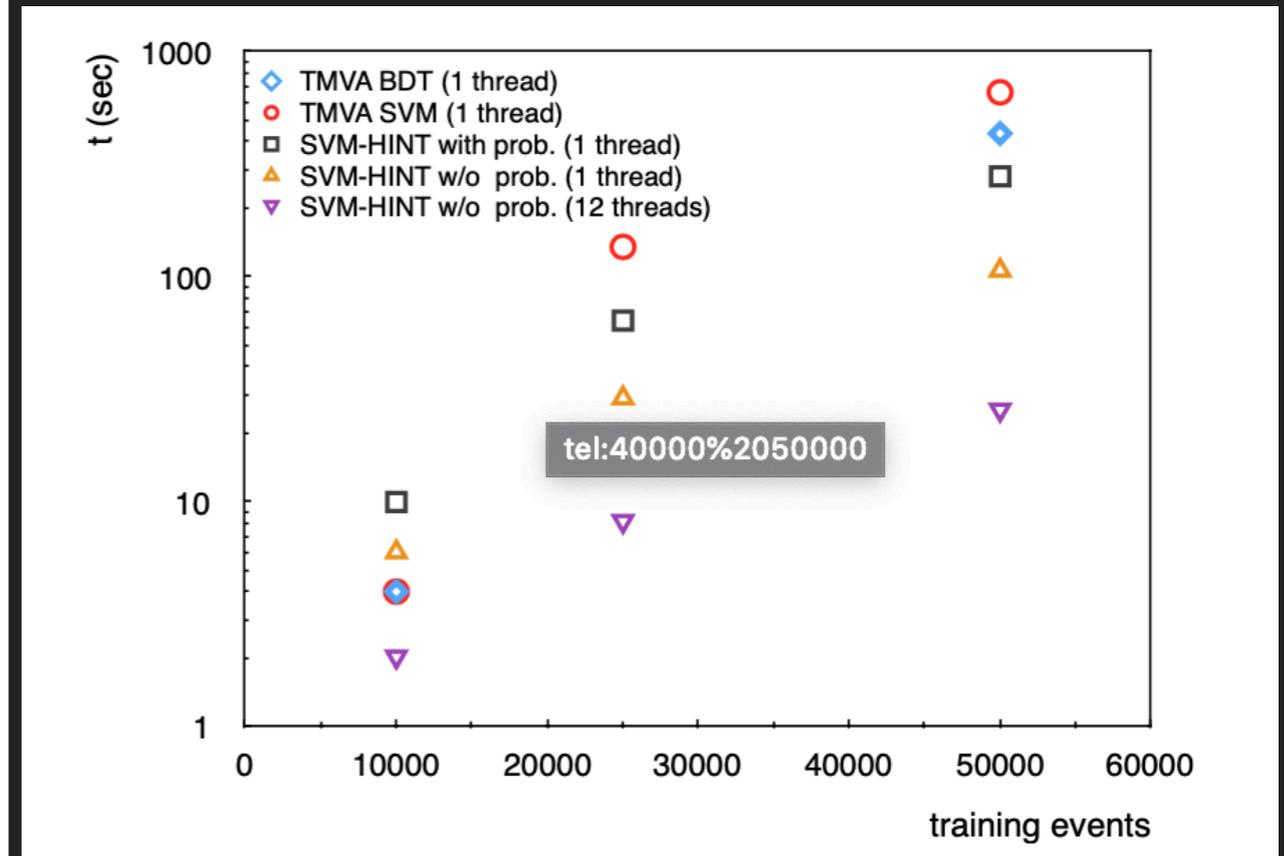
- ▶ Separación de datos por hiper-planos
- ▶ Usar Funciones de Kernel para encontrar un Support Vector Classifier, por validación cruzada



CLASIFICACIÓN BINARIA DE SEÑAL SUPER-SIMÉTRICA VS BACKGROUND DEL MODELO ESTÁNDAR

	Variable	Set 1	Set 2	Set 3	Set 4
low-level	$p_{T,l}$	•	•		
	η_l	•	•		
	$p_{T,jet(1,2,3,4)}$	•	•		
	$\eta_{jet(1,2,3,4)}$	•	•		
	$p_{T,bjet1}$	•	•		
	η_{bjet1}	•	•		
	n_{jet}	•	•		
	n_{bjet}	•	•		
	\cancel{E}_T	•	•		•
	H_T	•	•		•
high-level	m_T	•		•	•
	m_{T2}^W	•		•	•
	$\Delta\phi(W, l)$	•		•	
	$m(l, b)$	•		•	
	Centrality	•		•	
	Y	•		•	
	H_T -ratio	•		•	
	$\Delta r_{\min}(l, b)$	•		•	
	$\Delta\phi_{\min}(j_{1,2}, \cancel{E}_T)$	•		•	

(Sahin et. al. 2016)

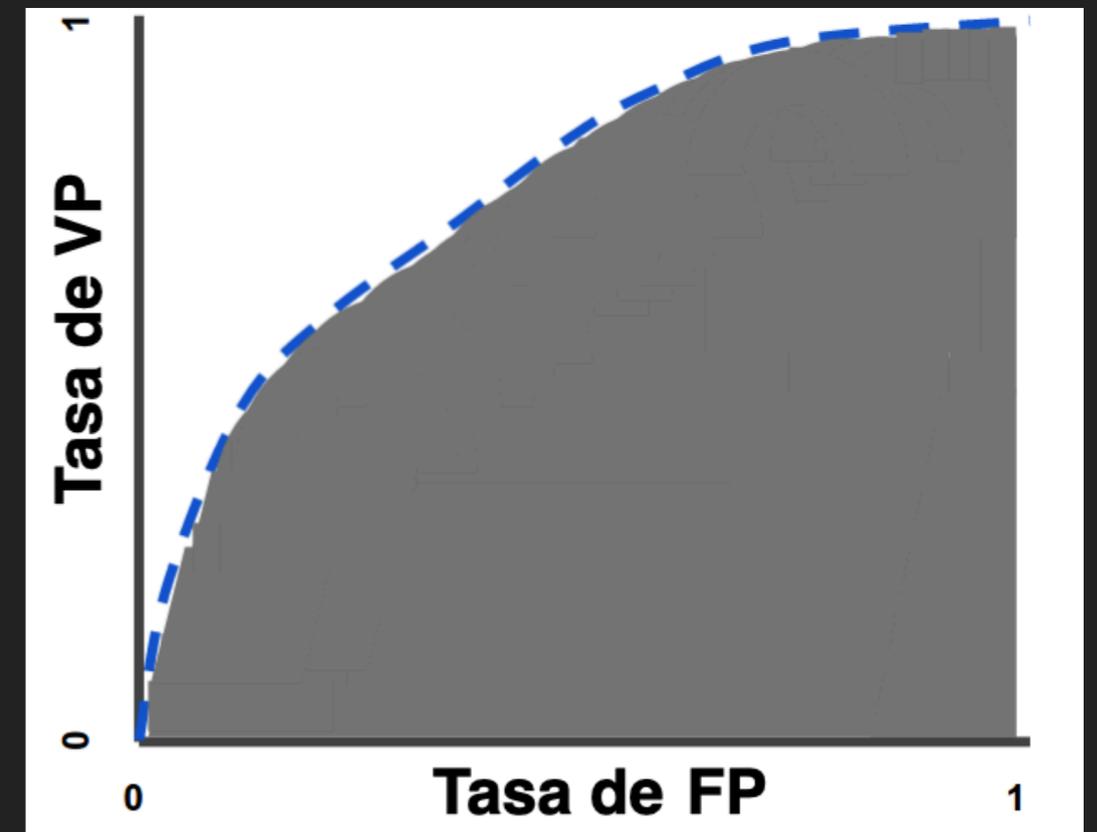


MEDIDA DE RENDIMIENTO

		Valor en la realidad		total
		p	n	
Predicción outcome	p'	Verdaderos Positivos	Falsos Positivos	P'
	n'	Falsos Negativos	Verdaderos Negativos	N'
total		P	N	

MEDIDA DE RENDIMIENTO

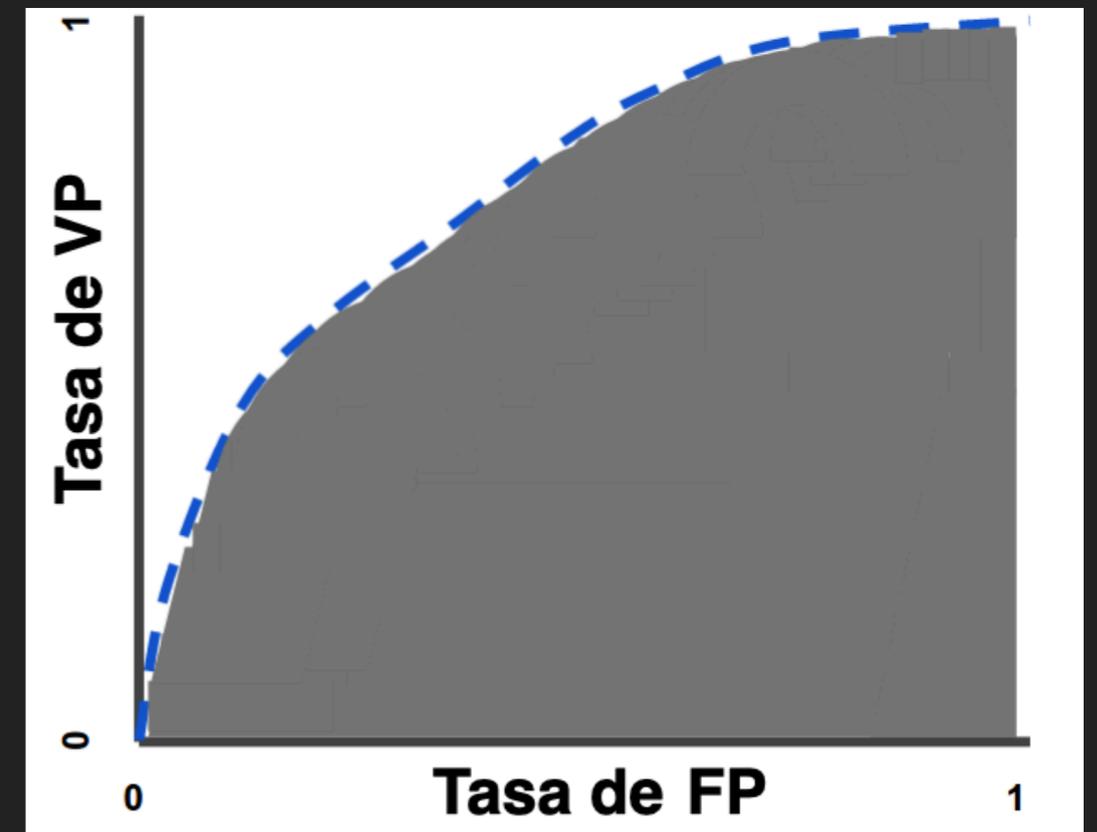
		Valor en la realidad		total
		p	n	
Predicción outcome	p'	Verdaderos Positivos	Falsos Positivos	P'
	n'	Falsos Negativos	Verdaderos Negativos	N'
total		P	N	



MEDIDA DE RENDIMIENTO

- ▶ Curva ROC (Característica Operativa del Receptor)

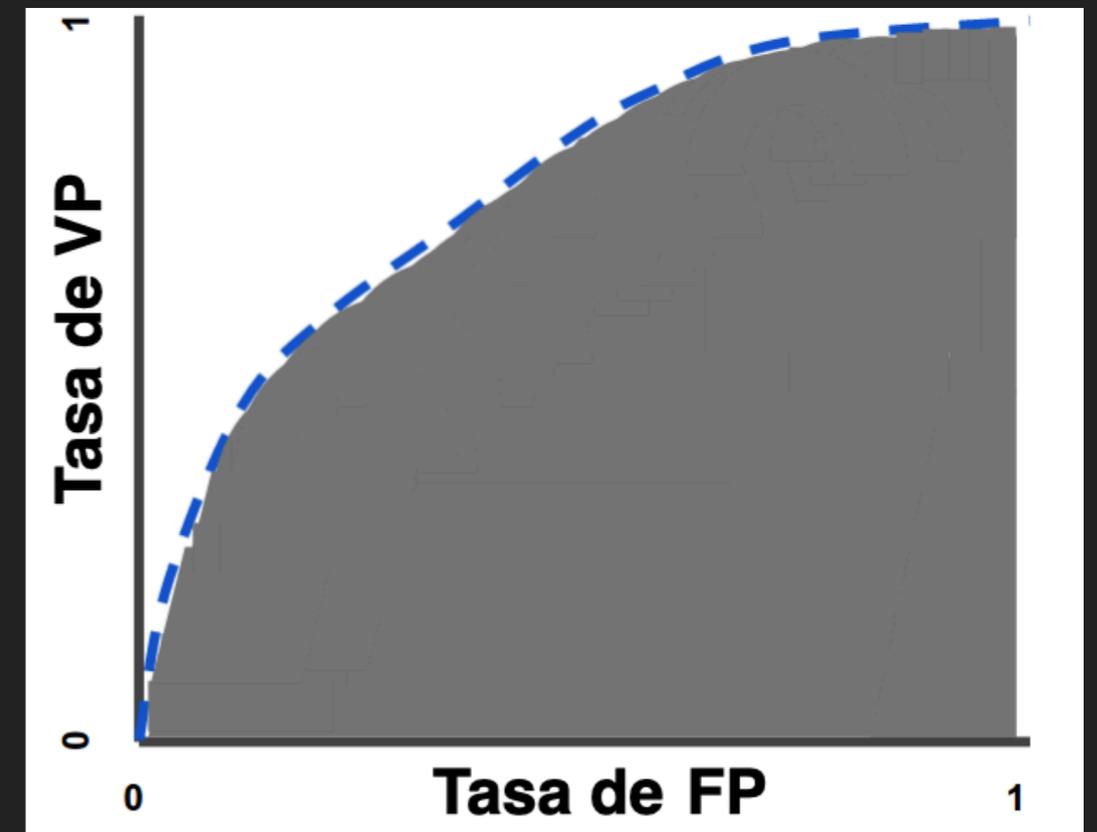
		Valor en la realidad		total
		p	n	
Predicción outcome	p'	Verdaderos Positivos	Falsos Positivos	P'
	n'	Falsos Negativos	Verdaderos Negativos	N'
total		P	N	



MEDIDA DE RENDIMIENTO

- ▶ Curva ROC (Característica Operativa del Receptor)
- ▶ Normalmente: True Positive Rate vs False Positive Rate para diferentes umbrales

		Valor en la realidad		total
		p	n	
Predicción outcome	p'	Verdaderos Positivos	Falsos Positivos	P'
	n'	Falsos Negativos	Verdaderos Negativos	N'
total		P	N	

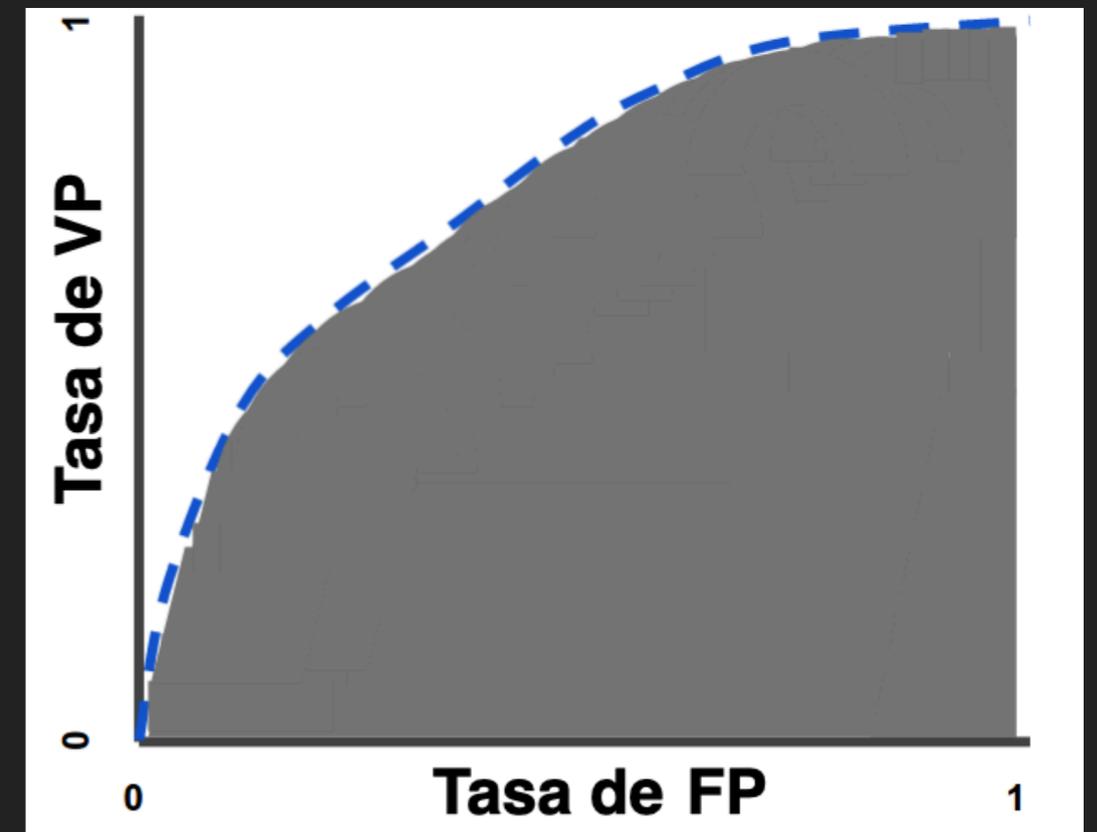


MEDIDA DE RENDIMIENTO

- ▶ Curva ROC (Característica Operativa del Receptor)
- ▶ Normalmente: True Positive Rate vs False Positive Rate para diferentes umbrales

$$\frac{VP}{P} \text{ vs } \frac{FP}{N}$$

		Valor en la realidad		total
		<i>p</i>	<i>n</i>	
Predicción outcome	<i>p'</i>	Verdaderos Positivos	Falsos Positivos	<i>P'</i>
	<i>n'</i>	Falsos Negativos	Verdaderos Negativos	<i>N'</i>
total		<i>P</i>	<i>N</i>	



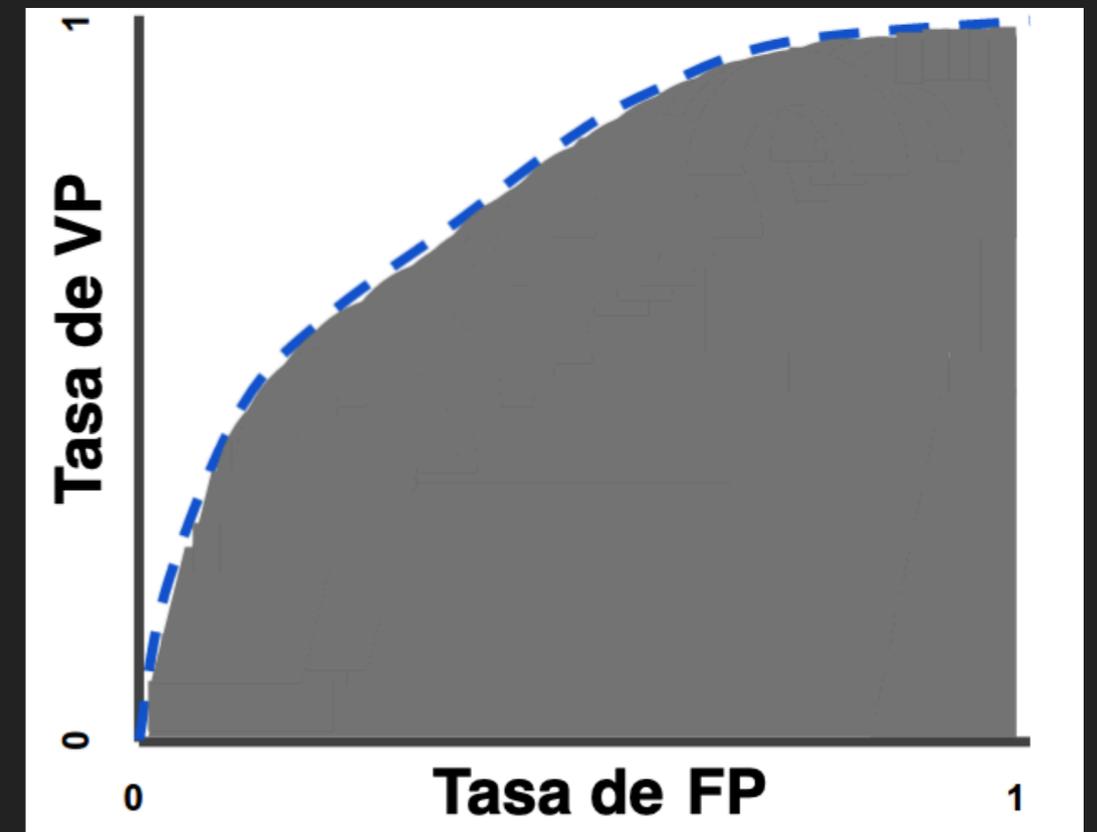
MEDIDA DE RENDIMIENTO

- ▶ Curva ROC (Característica Operativa del Receptor)
- ▶ Normalmente: True Positive Rate vs False Positive Rate para diferentes umbrales

$$\frac{VP}{P} \text{ vs } \frac{FP}{N}$$

- ▶ En HEP: Rechazo de Background vs Eficiencia de señal

		Valor en la realidad		total
		<i>p</i>	<i>n</i>	
Predicción outcome	<i>p'</i>	Verdaderos Positivos	Falsos Positivos	<i>P'</i>
	<i>n'</i>	Falsos Negativos	Verdaderos Negativos	<i>N'</i>
total		<i>P</i>	<i>N</i>	



MEDIDA DE RENDIMIENTO

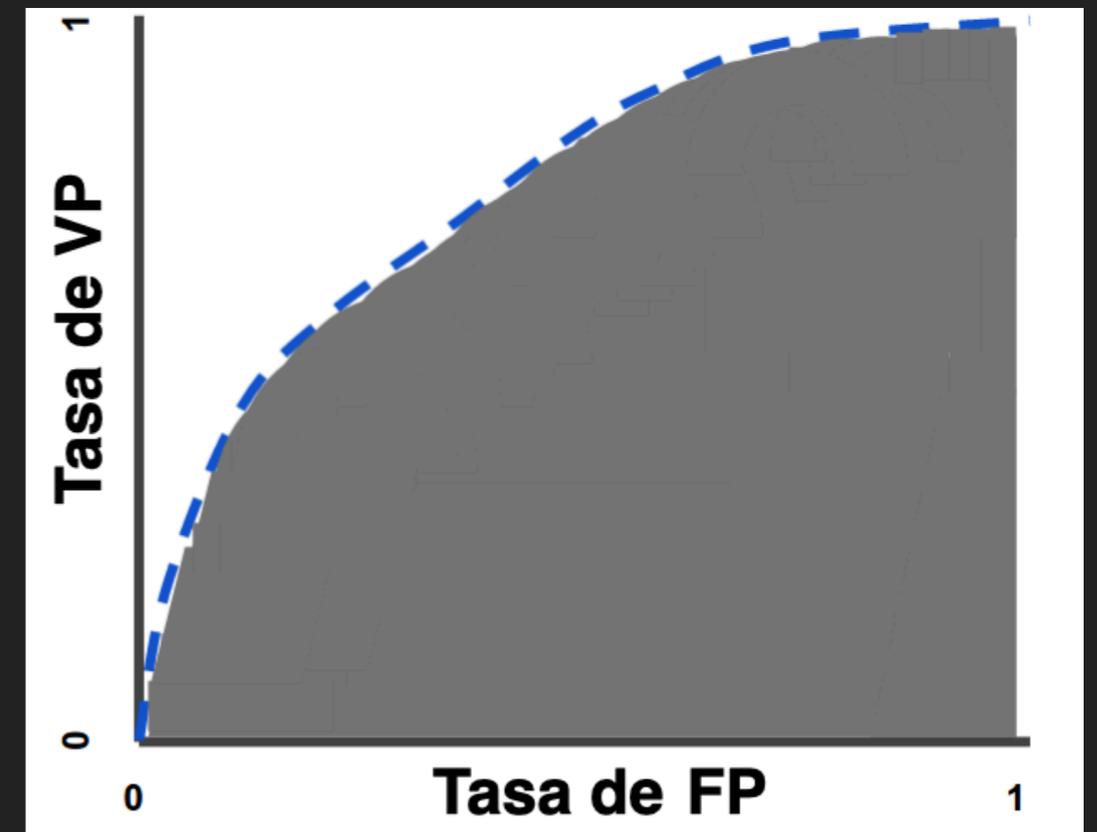
- ▶ Curva ROC (Característica Operativa del Receptor)
- ▶ Normalmente: True Positive Rate vs False Positive Rate para diferentes umbrales

$$\frac{VP}{P} \text{ vs } \frac{FP}{N}$$

- ▶ En HEP: Rechazo de Background vs Eficiencia de señal

$$1 - \frac{FP}{N} \text{ vs } \frac{VP}{P}$$

		Valor en la realidad		total
		<i>p</i>	<i>n</i>	
Predicción outcome	<i>p'</i>	Verdaderos Positivos	Falsos Positivos	<i>P'</i>
	<i>n'</i>	Falsos Negativos	Verdaderos Negativos	<i>N'</i>
total		<i>P</i>	<i>N</i>	



MEDIDA DE RENDIMIENTO

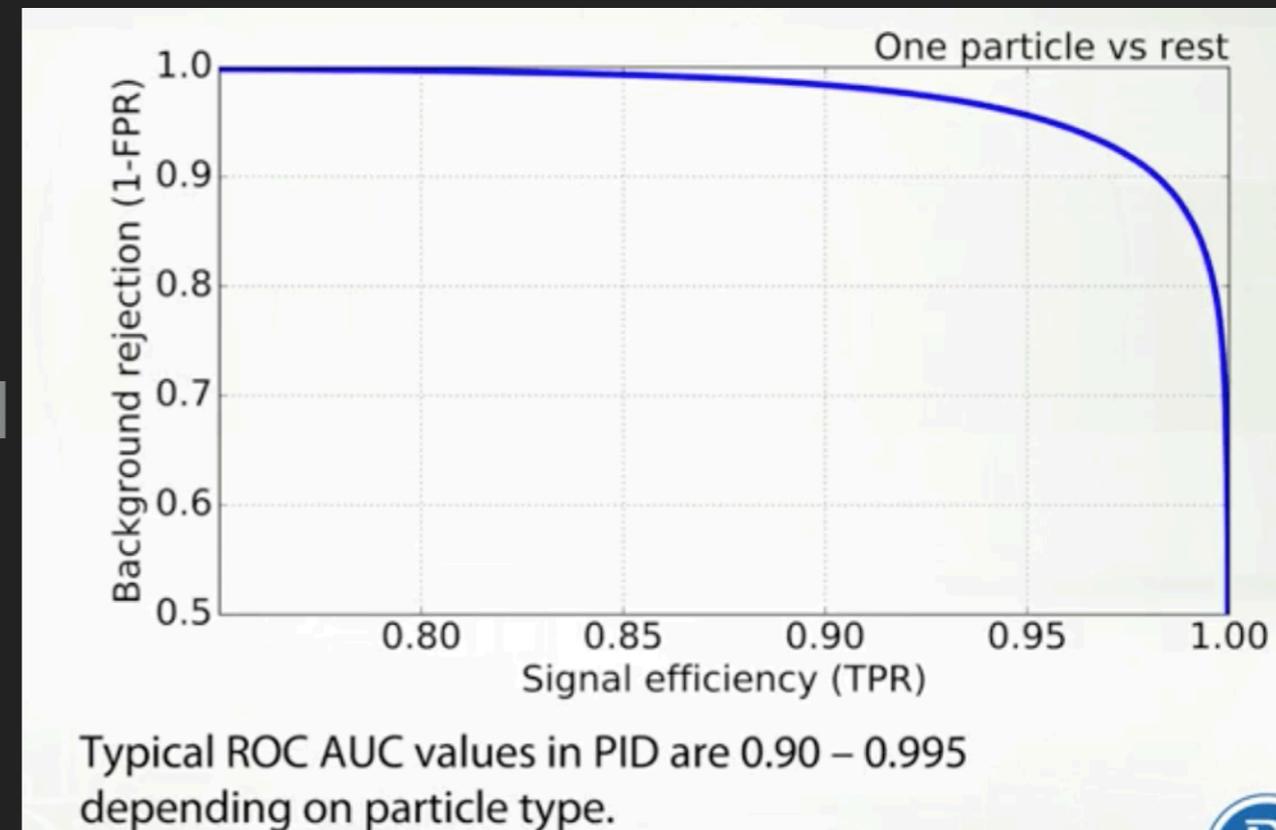
- ▶ Curva ROC (Característica Operativa del Receptor)
- ▶ Normalmente: True Positive Rate vs False Positive Rate para diferentes umbrales

$$\frac{VP}{P} \text{ vs } \frac{FP}{N}$$

- ▶ En HEP: Rechazo de Background vs Eficiencia de señal

$$1 - \frac{FP}{N} \text{ vs } \frac{VP}{P}$$

		Valor en la realidad		total
		<i>p</i>	<i>n</i>	
Predicción outcome	<i>p'</i>	Verdaderos Positivos	Falsos Positivos	<i>P'</i>
	<i>n'</i>	Falsos Negativos	Verdaderos Negativos	<i>N'</i>
total		<i>P</i>	<i>N</i>	



MEDIDA DE RENDIMIENTO PARA FÍSICA DE PARTÍCULAS

(Sahin et. al. 2016)

MEDIDA DE RENDIMIENTO PARA FÍSICA DE PARTÍCULAS

- ▶ Optimización de parámetros para maximizar significación estadística

(Sahin et. al. 2016)

MEDIDA DE RENDIMIENTO PARA FÍSICA DE PARTÍCULAS

- ▶ Optimización de parámetros para maximizar significación estadística
- ▶ Estimación de significación de descubrimiento por estimada de Asimov

(Sahin et. al. 2016)

MEDIDA DE RENDIMIENTO PARA FÍSICA DE PARTÍCULAS

- ▶ Optimización de parámetros para maximizar significación estadística
- ▶ Estimación de significación de descubrimiento por estimada de Asimov

$$Z_A = \left[2 \left((s + b) \ln \left[\frac{(s + b)(b + \sigma_b^2)}{b^2 + (s + b)\sigma_b^2} \right] - \frac{b^2}{\sigma_b^2} \ln \left[1 + \frac{\sigma_b^2 s}{b(b + \sigma_b^2)} \right] \right) \right]^{1/2}$$

(Sahin et. al. 2016)

POR QUE LAS REDES NEURONALES?

POR QUE LAS REDES NEURONALES?

- ▶ Redes neuronales como aprendizaje representativo

POR QUE LAS REDES NEURONALES?

- ▶ Redes neuronales como aprendizaje representativo
- ▶ Pueden aprovechar grandes cantidades de datos

POR QUE LAS REDES NEURONALES?

- ▶ Redes neuronales como aprendizaje representativo
- ▶ Pueden aprovechar grandes cantidades de datos
 - ▶ expresividad & "entranabilidad"

POR QUE LAS REDES NEURONALES?

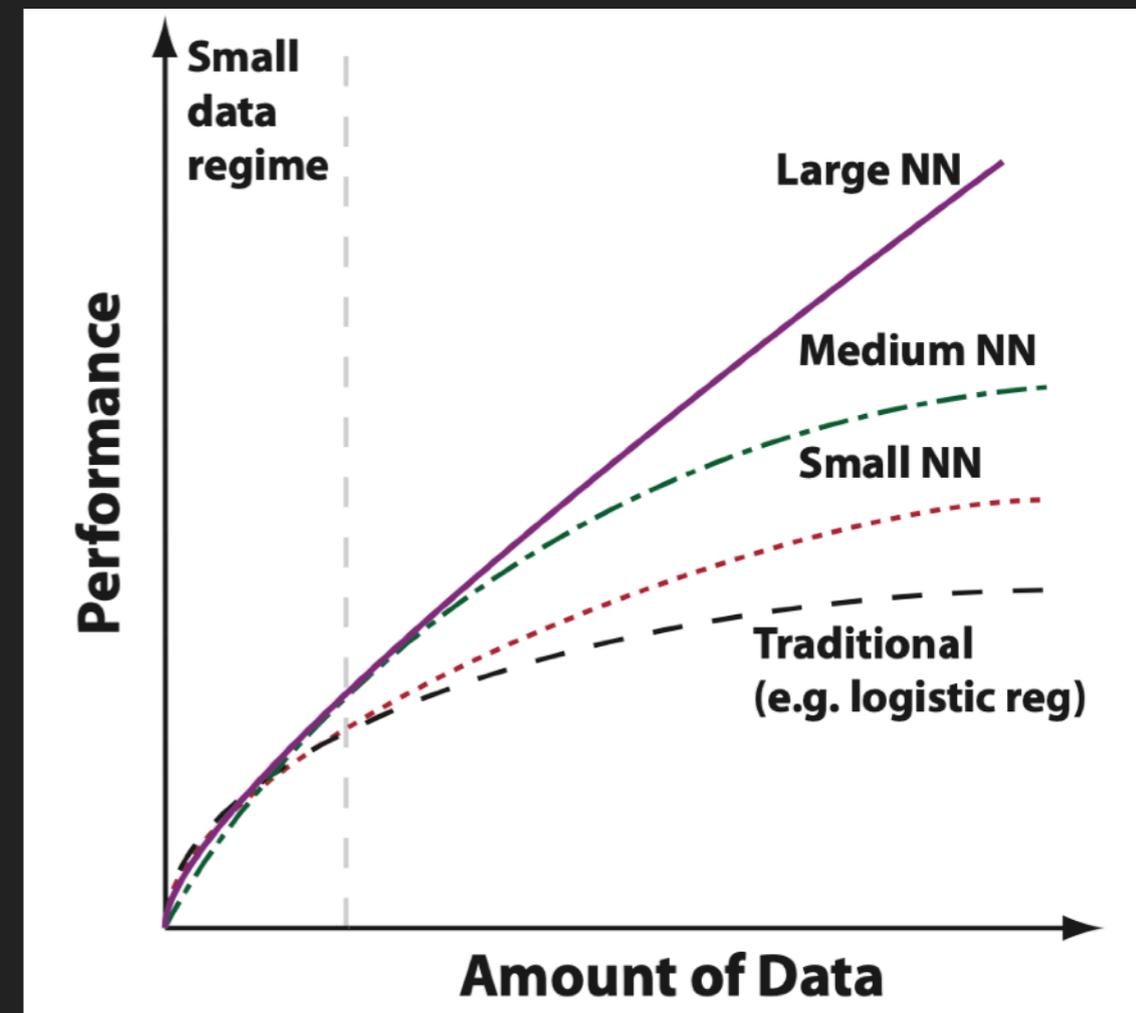
- ▶ Redes neuronales como aprendizaje representativo
- ▶ Pueden aprovechar grandes cantidades de datos
 - ▶ expresividad & "entranabilidad"
- ▶ Escalan bien con grandes cantidades de datos

POR QUE LAS REDES NEURONALES?

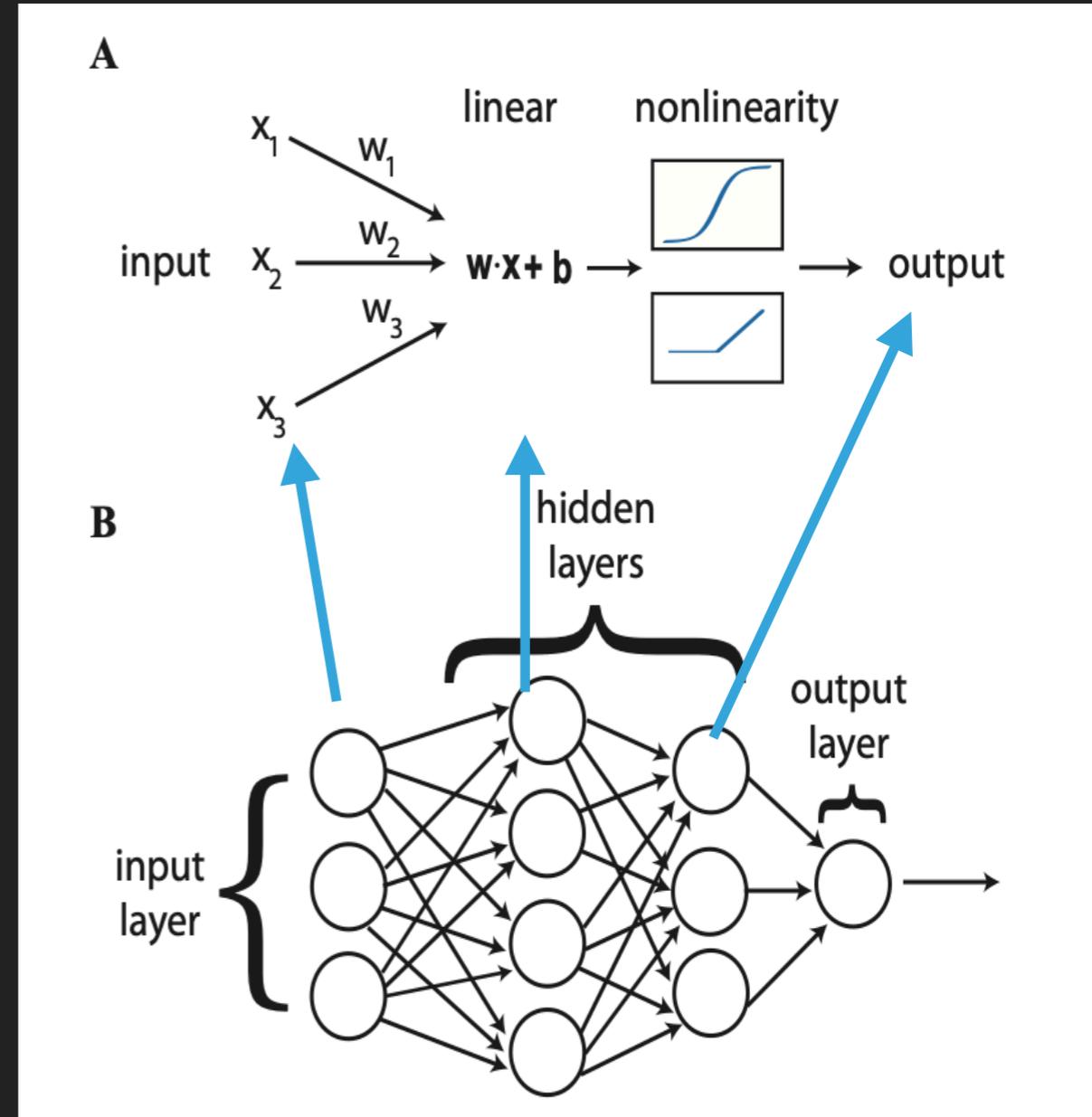
- ▶ Redes neuronales como aprendizaje representativo
- ▶ Pueden aprovechar grandes cantidades de datos
 - ▶ expresividad & "entranabilidad"
- ▶ Escalan bien con grandes cantidades de datos
 - ▶ GPU's, TPU's, Paralelización

POR QUE LAS REDES NEURONALES?

- ▶ Redes neuronales como aprendizaje representativo
- ▶ Pueden aprovechar grandes cantidades de datos
 - ▶ expresividad & "entranabilidad"
- ▶ Escalan bien con grandes cantidades de datos
 - ▶ GPU's, TPU's, Paralelización

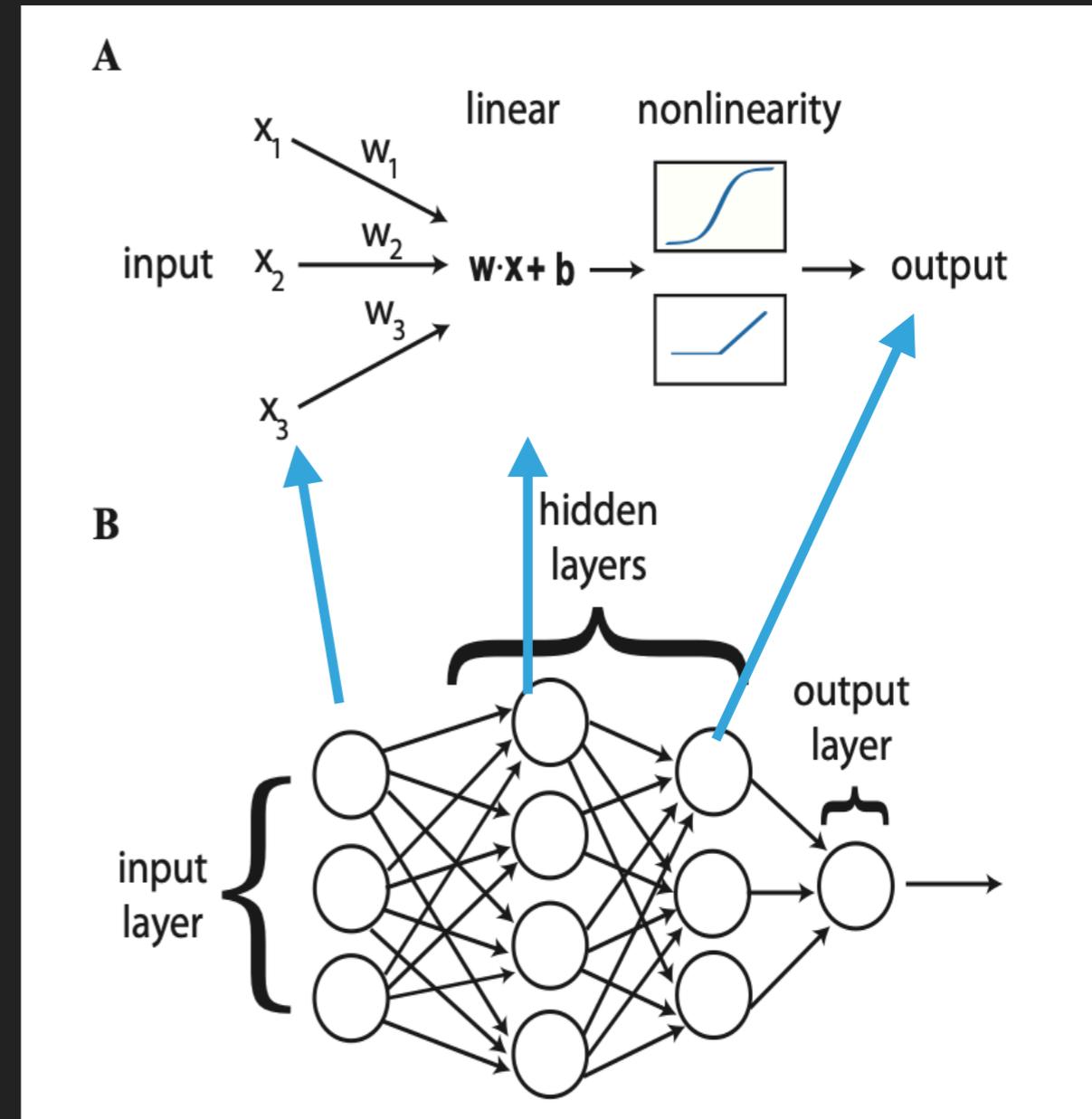


REDES NEURONALES



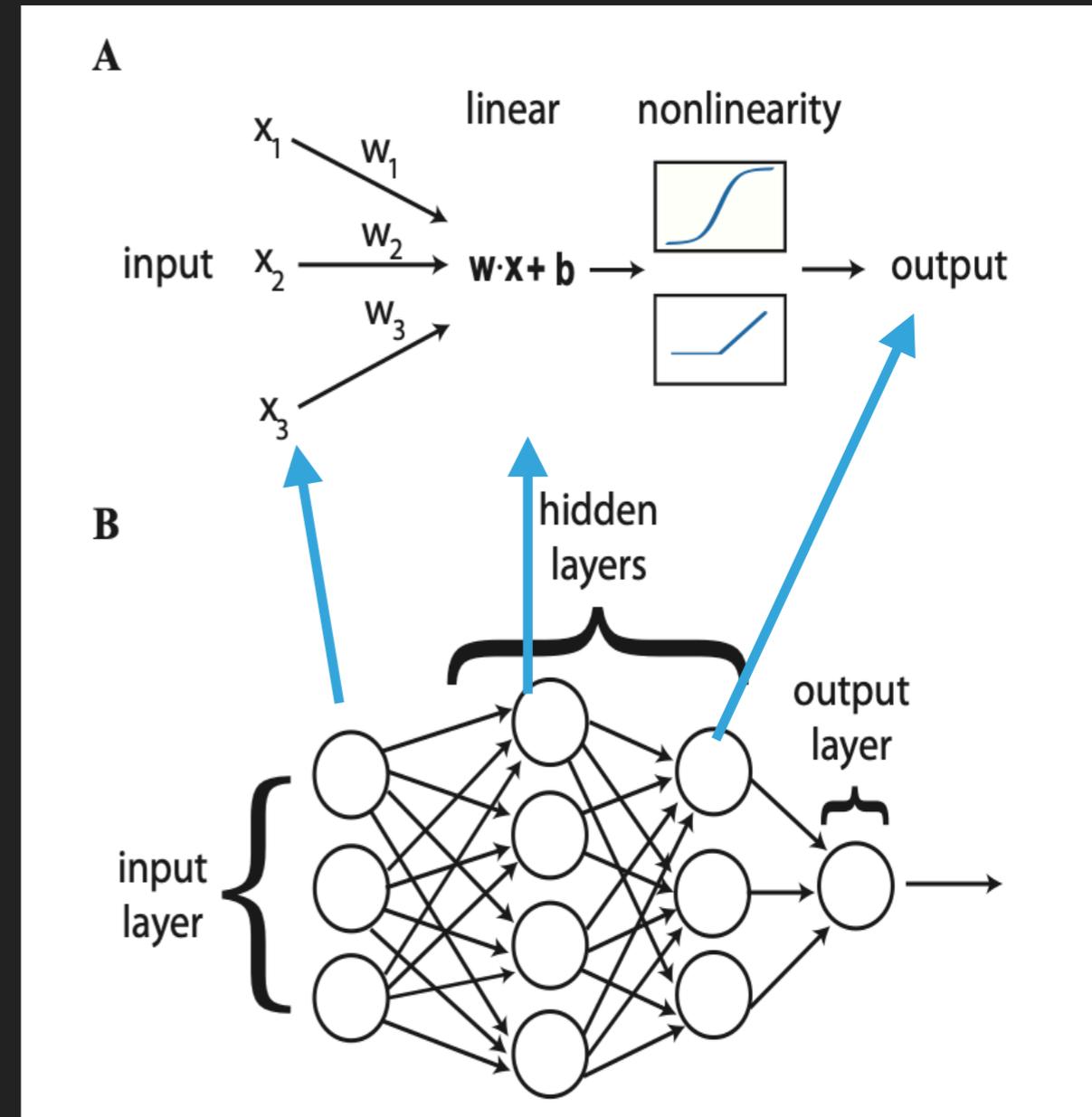
REDES NEURONALES

- ▶ Red de transformaciones lineales (seguidas por no-lineales) inicialmente en nuestros datos



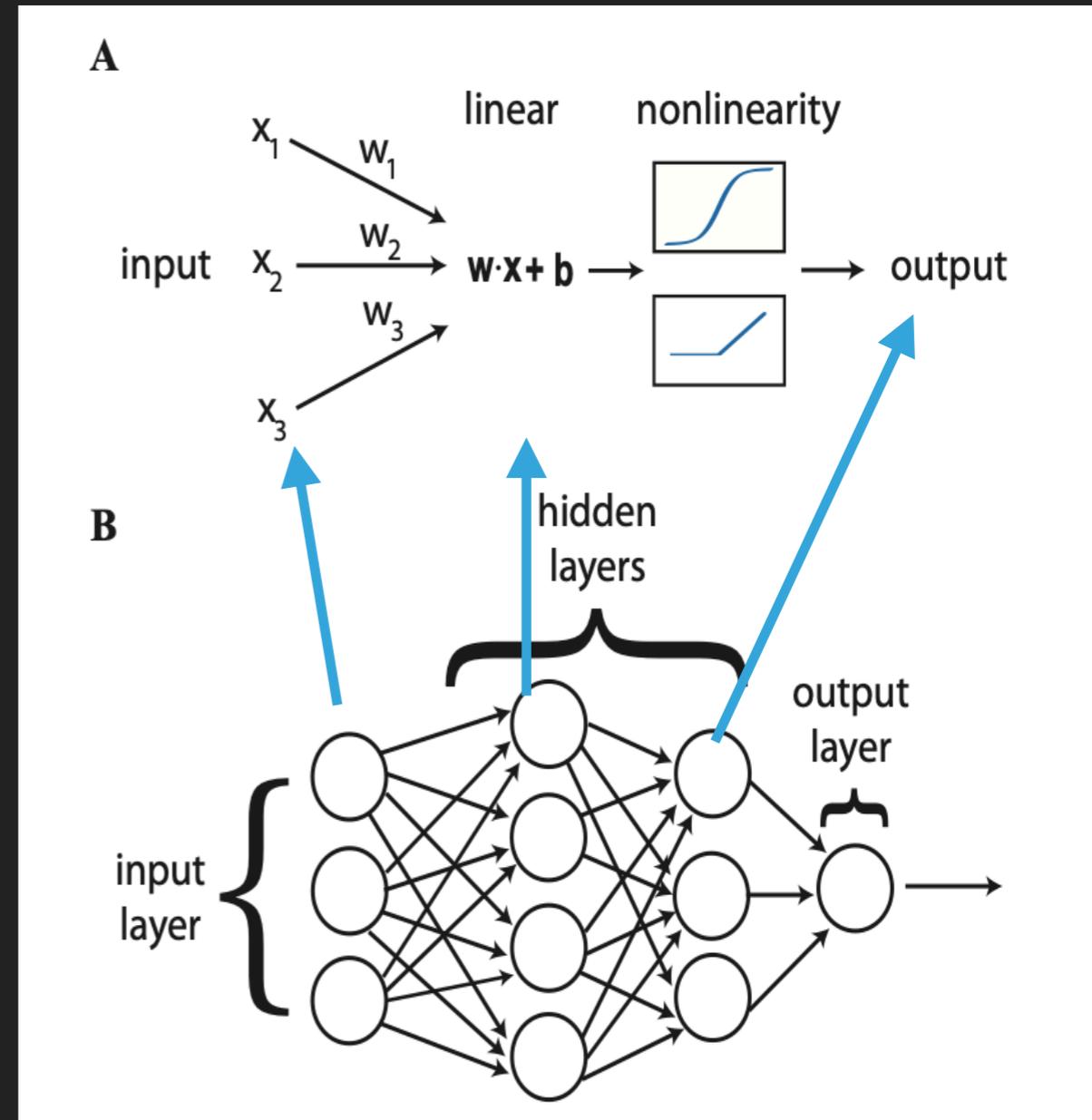
REDES NEURONALES

- ▶ Red de transformaciones lineales (seguidas por no-lineales) inicialmente en nuestros datos
- ▶ Redes neuronales profundas



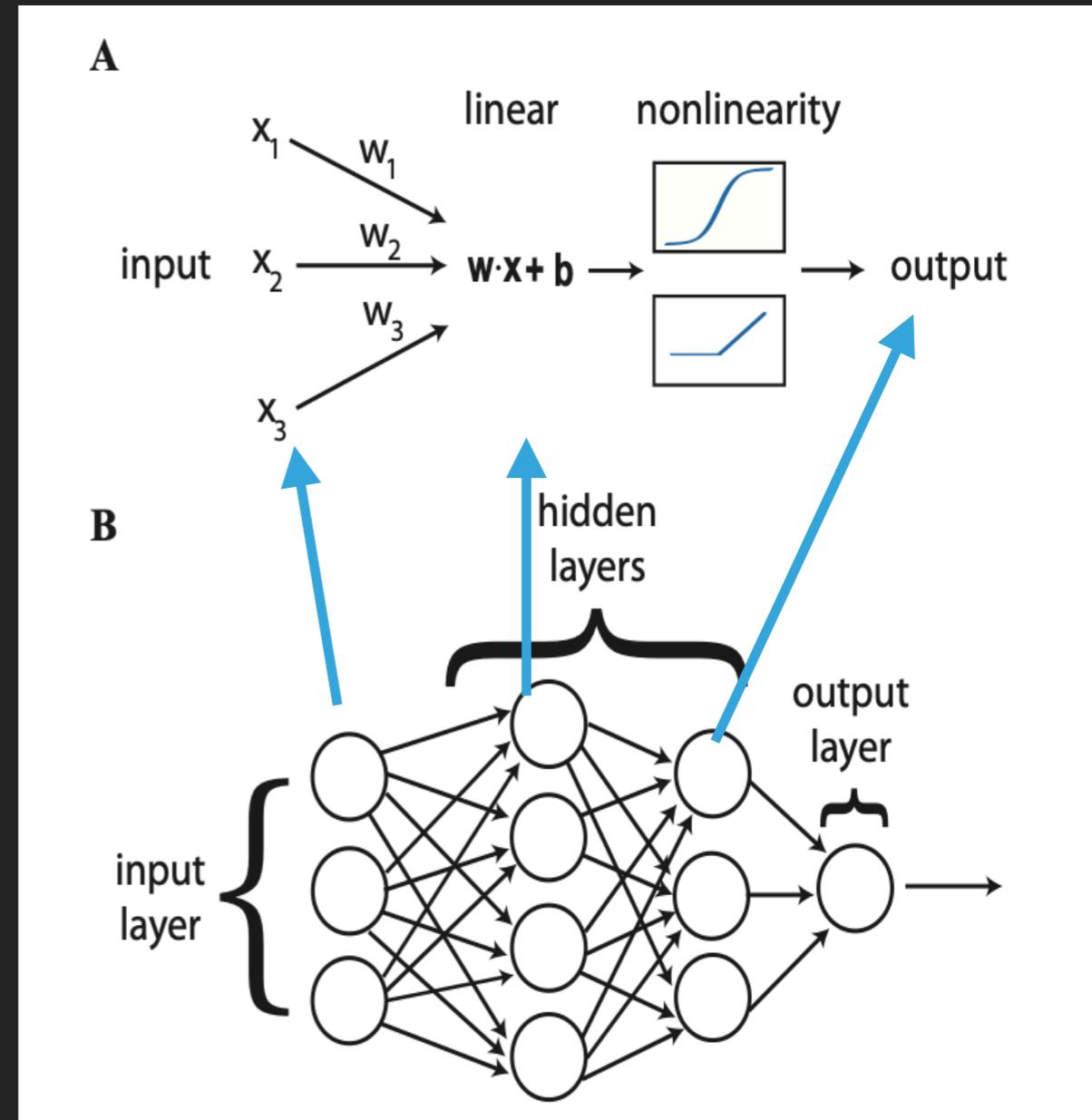
REDES NEURONALES

- ▶ Red de transformaciones lineales (seguidas por no-lineales) inicialmente en nuestros datos
- ▶ Redes neuronales profundas
 - ▶ capas ocultas/ hidden layers > 1 o 2



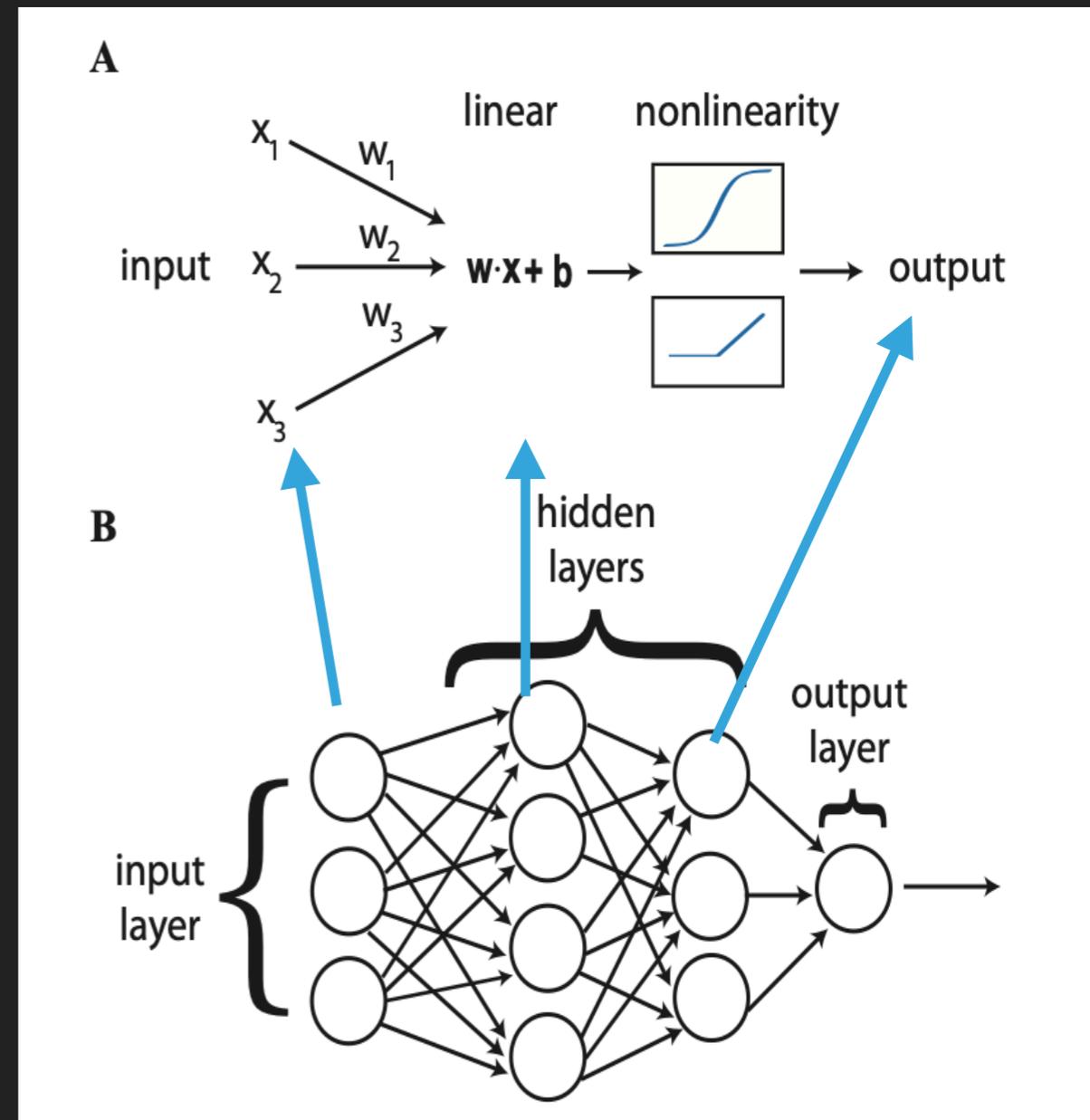
REDES NEURONALES

- ▶ Red de transformaciones lineales (seguidas por no-lineales) inicialmente en nuestros datos
- ▶ Redes neuronales profundas
 - ▶ capas ocultas/ hidden layers > 1 o 2
- ▶ Entrenamiento:



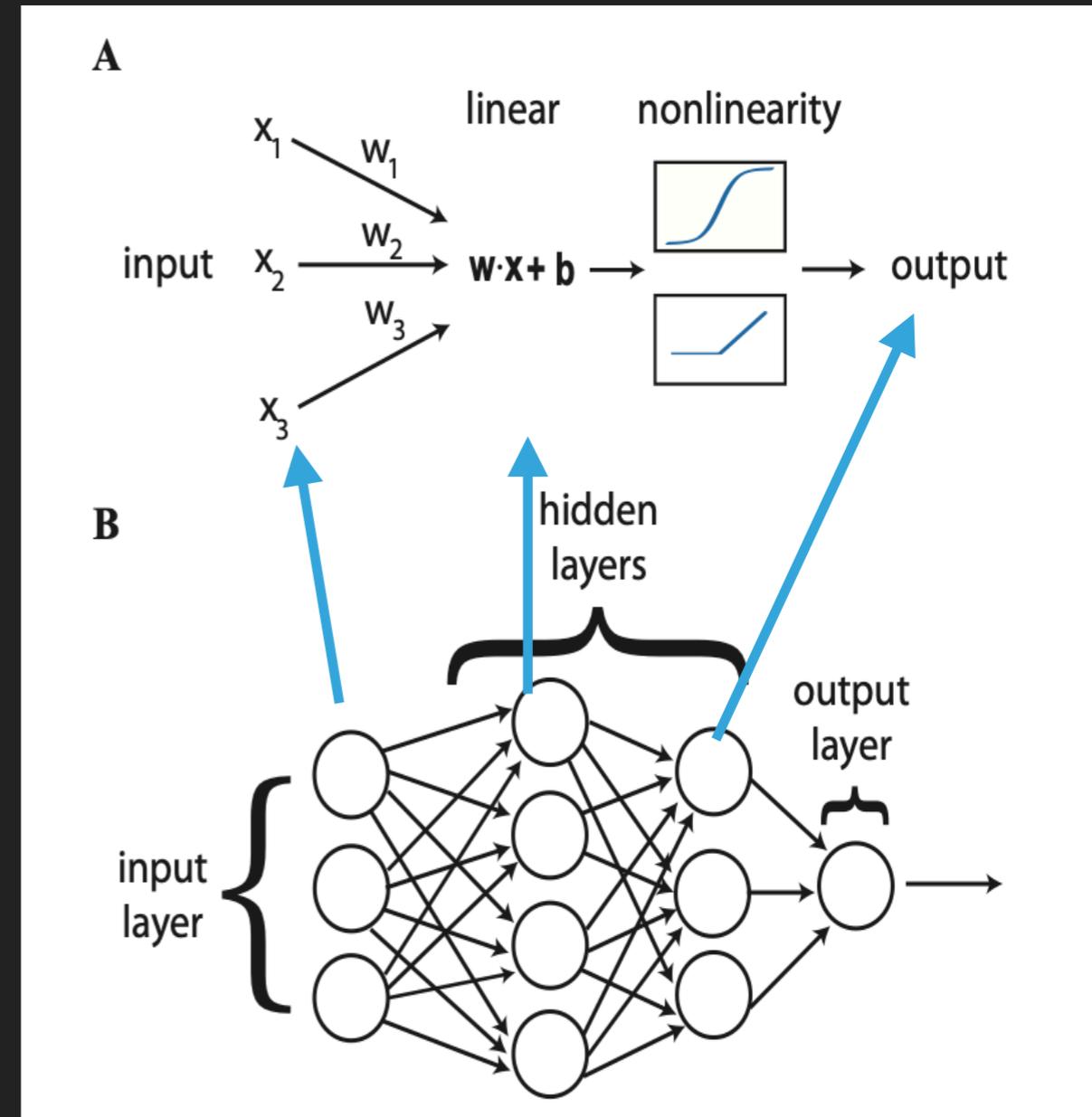
REDES NEURONALES

- ▶ Red de transformaciones lineales (seguidas por no-lineales) inicialmente en nuestros datos
- ▶ Redes neuronales profundas
 - ▶ capas ocultas/ hidden layers > 1 o 2
- ▶ Entrenamiento:
 - ▶ Descenso de gradiente

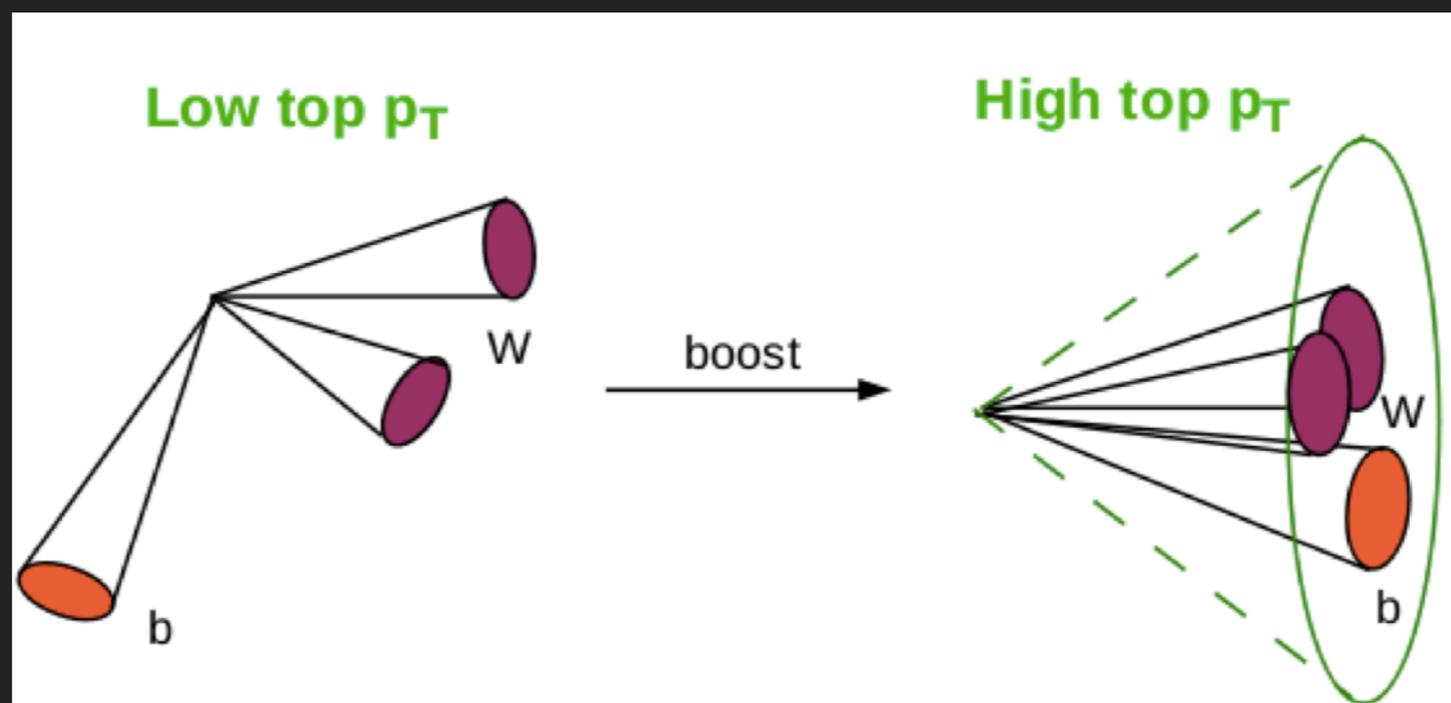


REDES NEURONALES

- ▶ Red de transformaciones lineales (seguidas por no-lineales) inicialmente en nuestros datos
- ▶ Redes neuronales profundas
 - ▶ capas ocultas/ hidden layers > 1 o 2
- ▶ Entrenamiento:
 - ▶ Descenso de gradiente
 - ▶ Backpropagation: calcular gradientes en capa l con gradientes de la capa $l + 1$

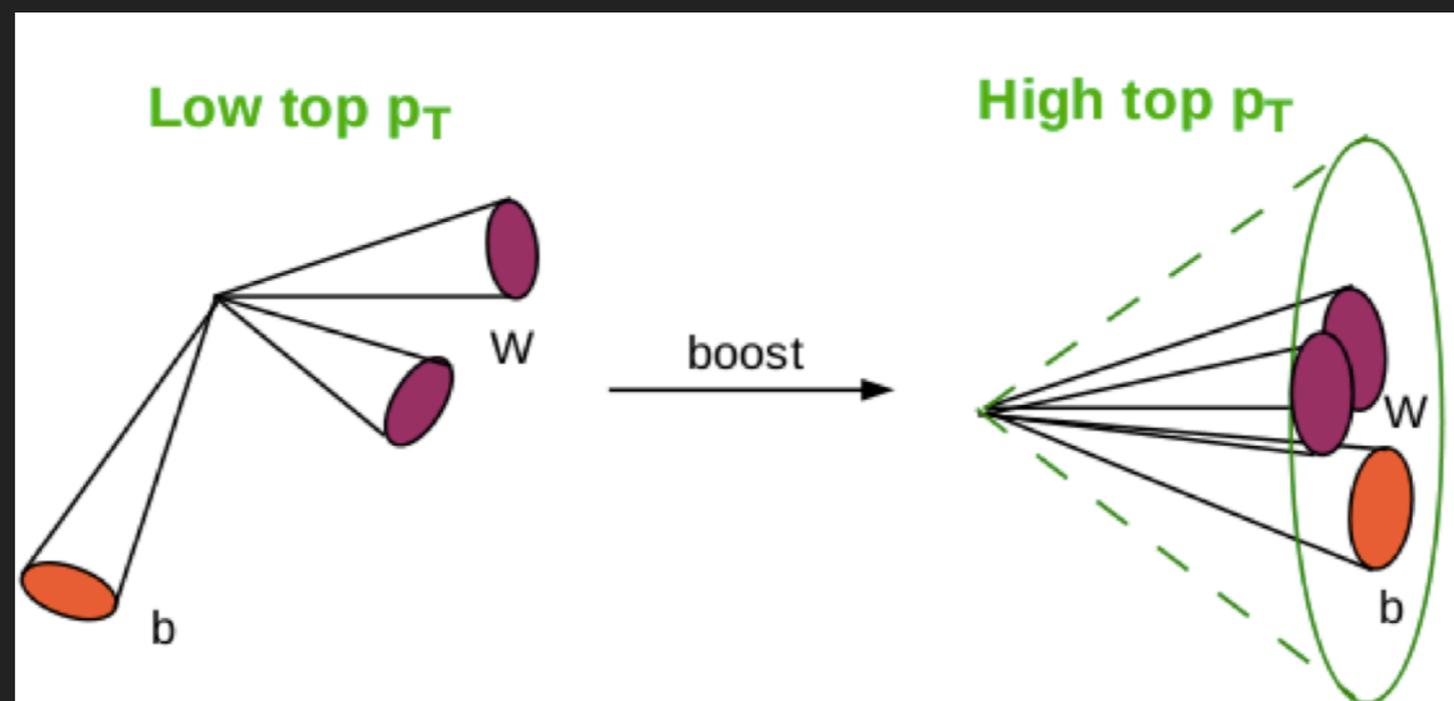


JET TAGGING



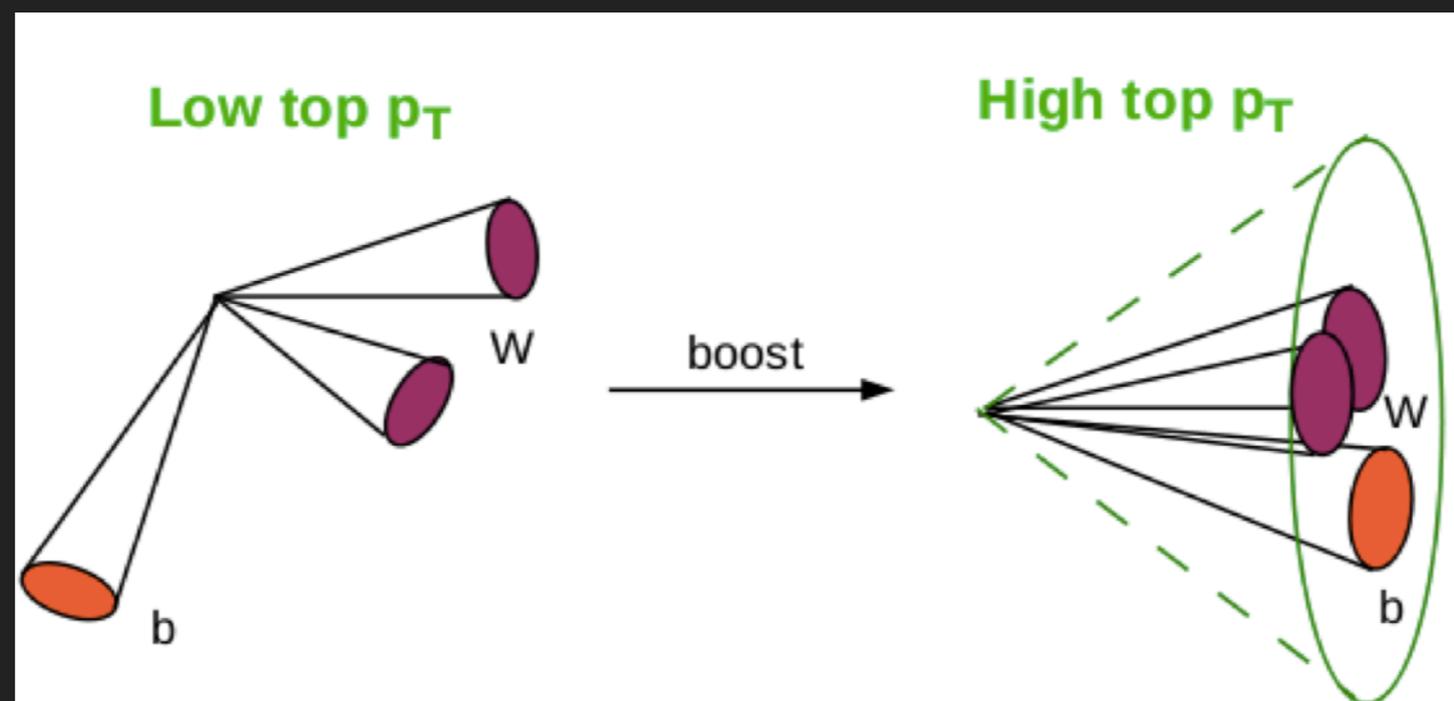
JET TAGGING

- ▶ Etiquetamiento de jets producidos por



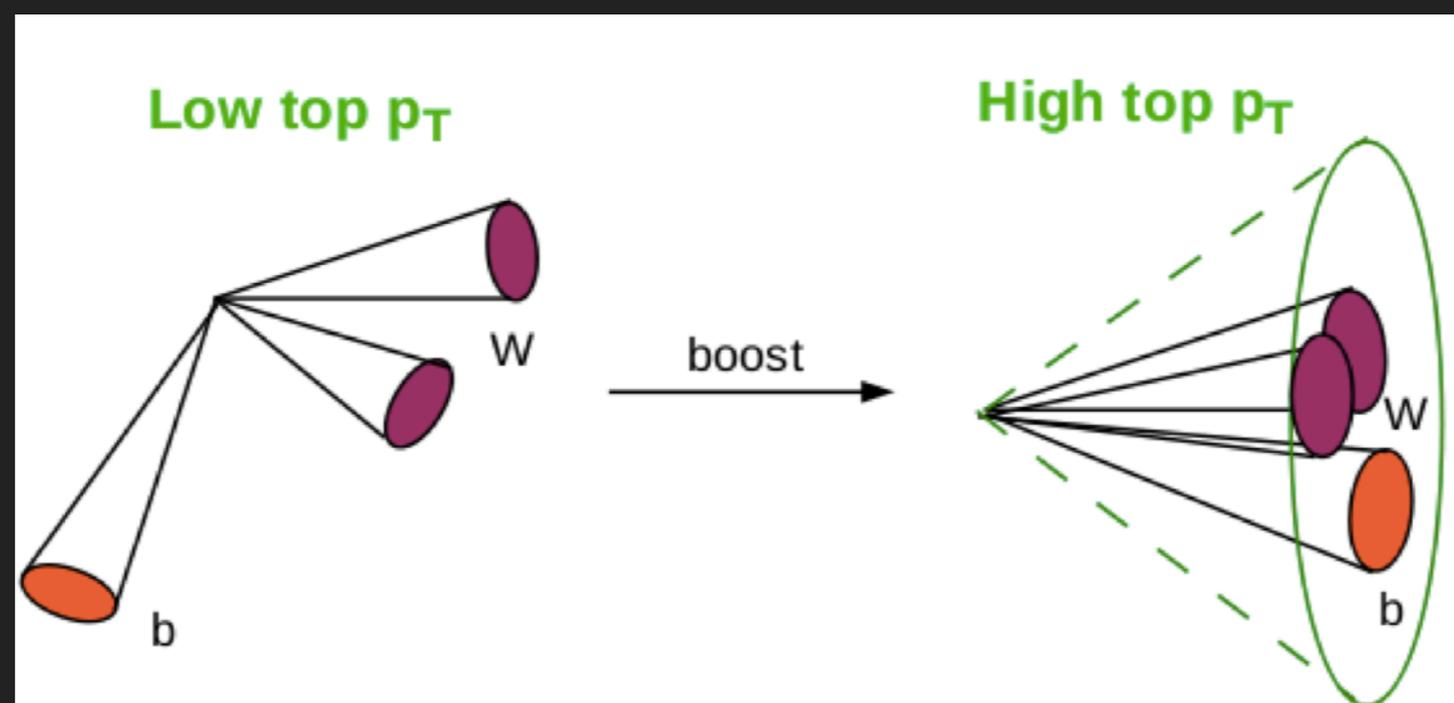
JET TAGGING

- ▶ Etiquetamiento de jets producidos por
 - ▶ b, c , quarks livianos \rightarrow etiquetamiento de sabor



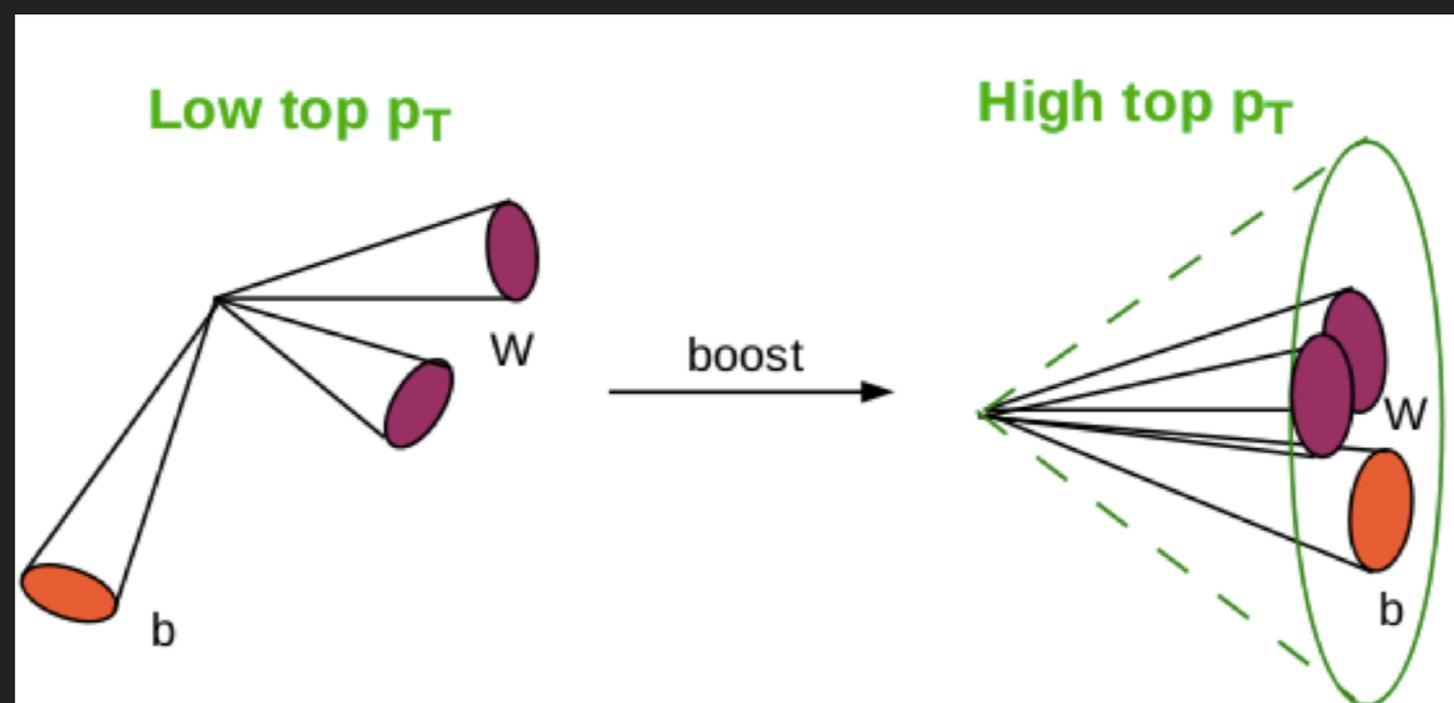
JET TAGGING

- ▶ Etiquetamiento de jets producidos por
 - ▶ b, c , quarks livianos \rightarrow etiquetamiento de sabor
 - ▶ W 's, Z , t , H \rightarrow etiquetamiento de subestructura



JET TAGGING

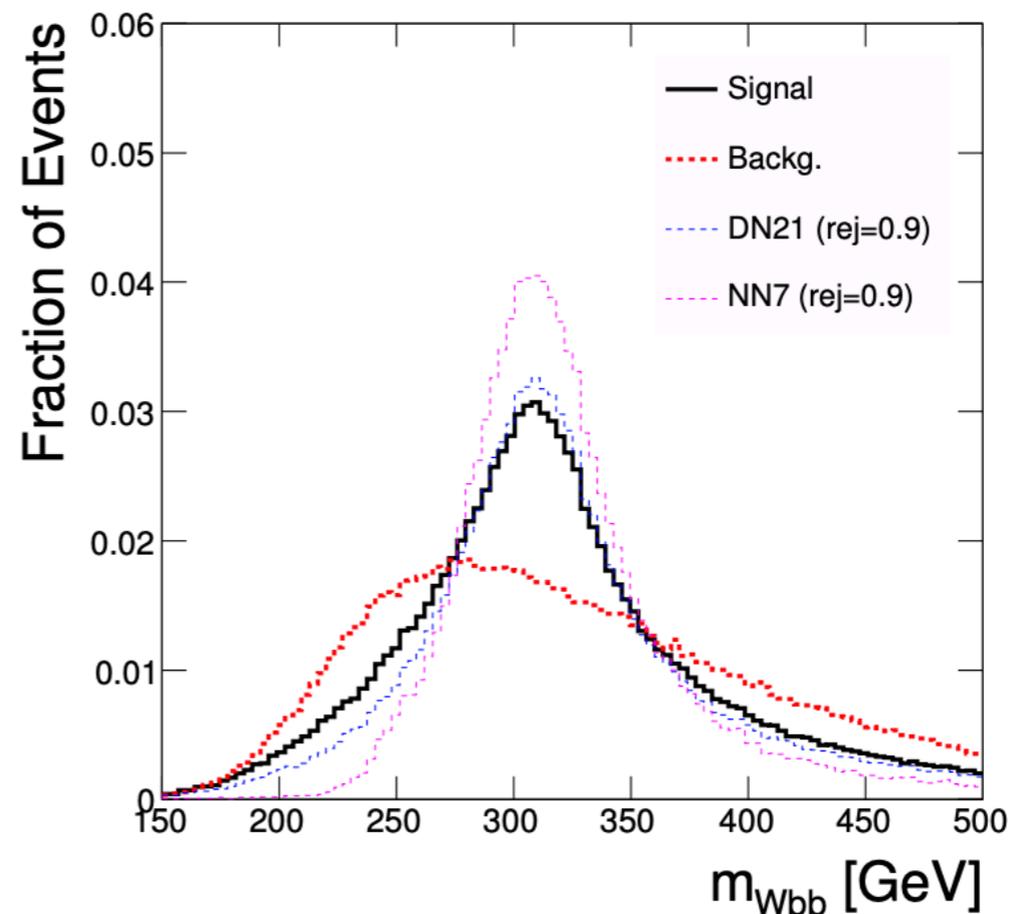
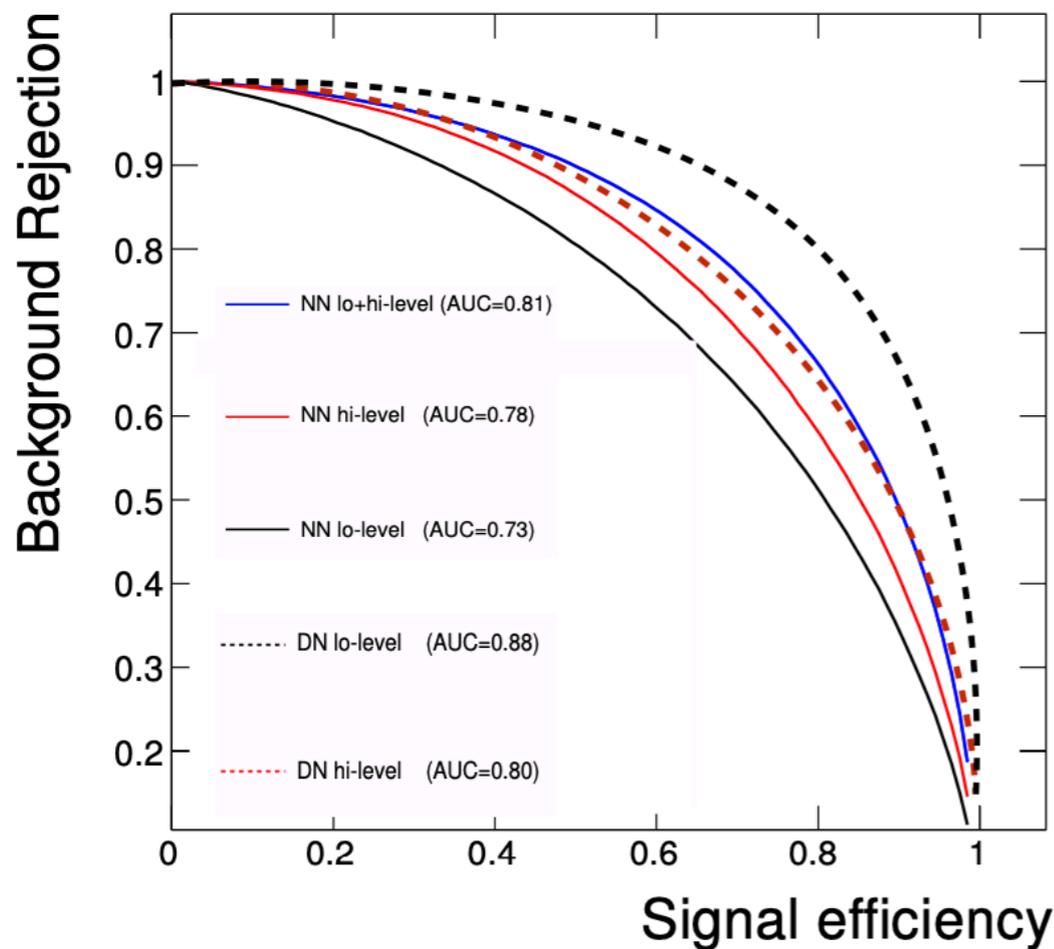
- ▶ Etiquetamiento de jets producidos por
 - ▶ b, c , quarks livianos \rightarrow etiquetamiento de sabor
 - ▶ W 's, Z , t , H \rightarrow etiquetamiento de subestructura
 - ▶ etiquetamiento de quark-gluon (QCD)



PRIMERAS IMPLEMENTACIONES DE DEEP LEARNING

(Baldi et al. 2014, Guest et al. 2019, Sahin et al. 2016)

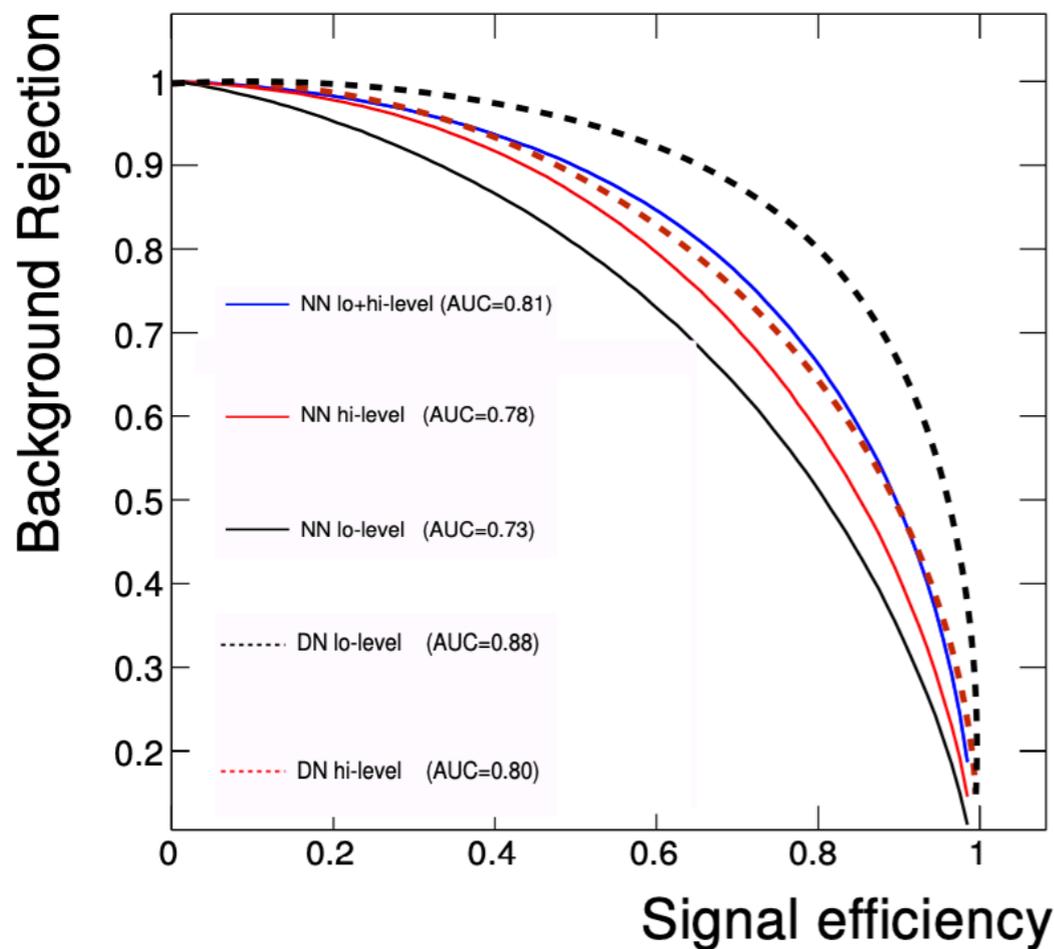
- Uso de redes profundas y poco profundas para discriminación de señales



PRIMERAS IMPLEMENTACIONES DE DEEP LEARNING

(Baldi et al. 2014, Guest et al. 2019, Sahin et al. 2016)

- Uso de redes profundas y poco profundas para discriminación de señales

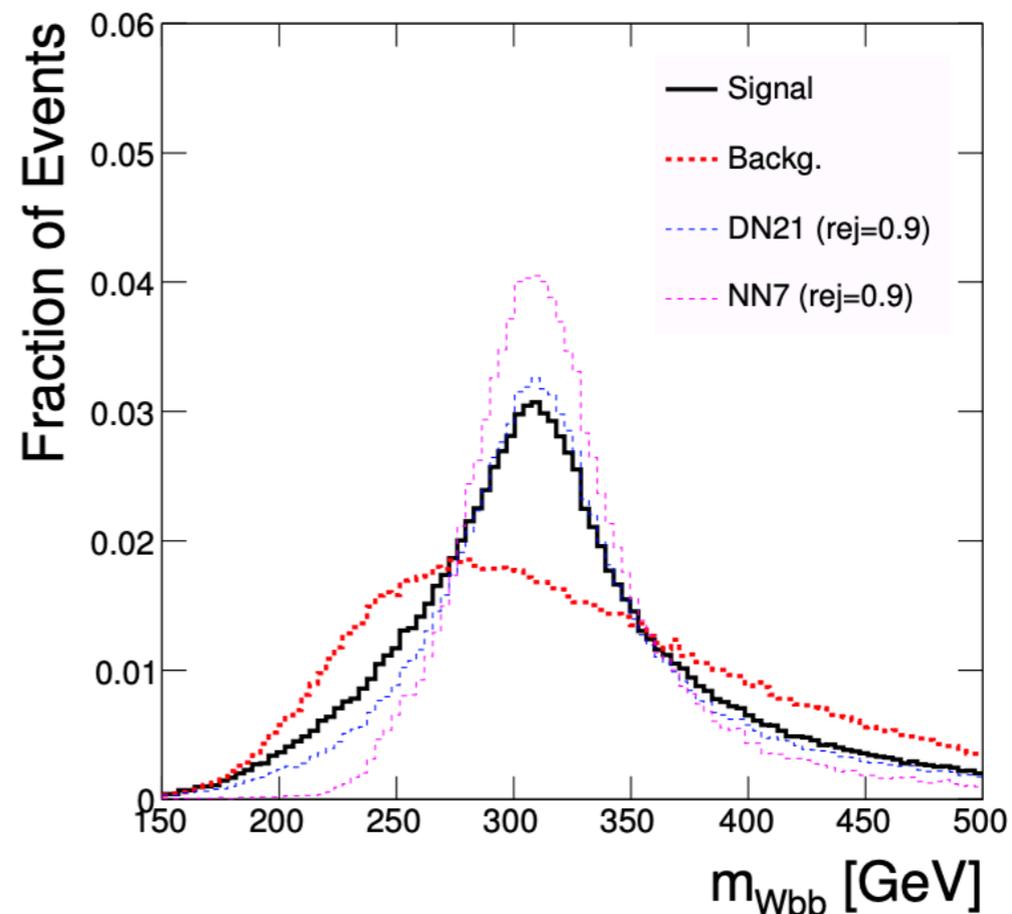
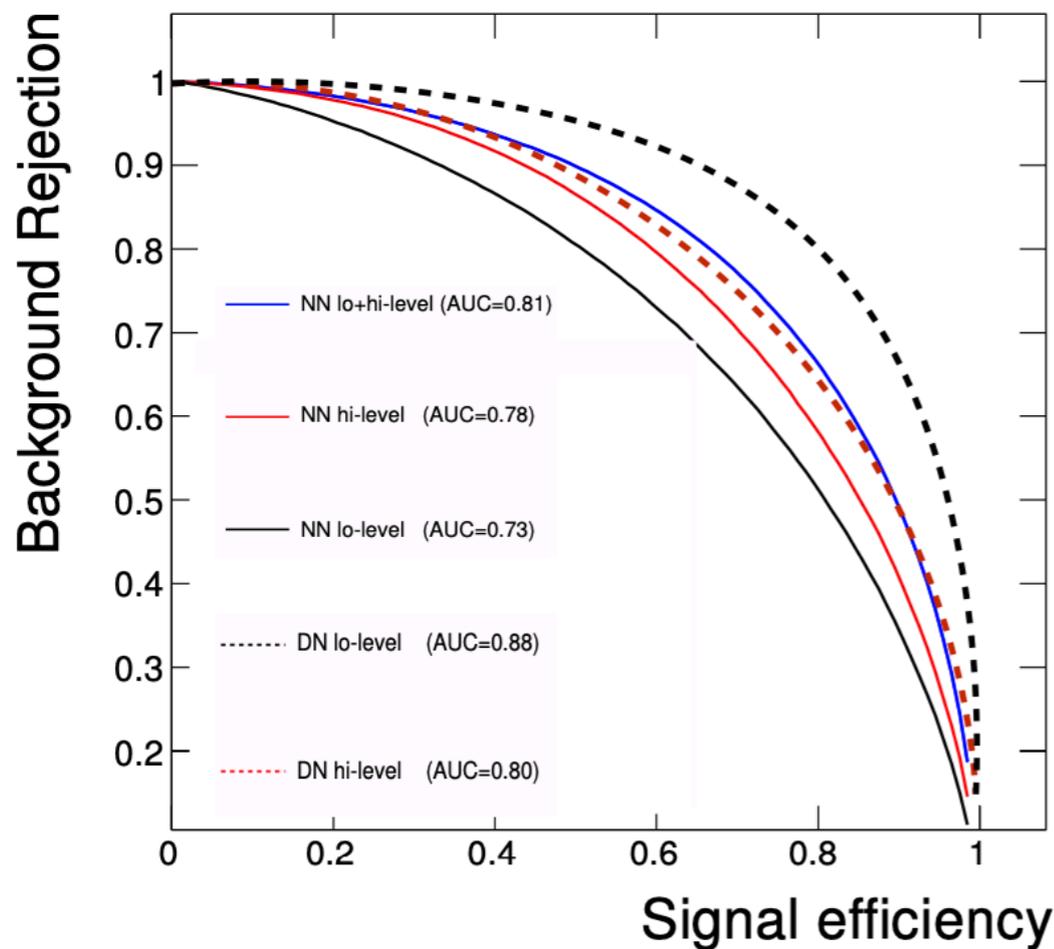


	Variable	Set 1	Set 2	Set 3	Set 4
low-level	$p_{T,l}$	•	•		
	η_l	•	•		
	$p_{T,jet(1,2,3,4)}$	•	•		
	$\eta_{jet(1,2,3,4)}$	•	•		
	$p_{T,bjet1}$	•	•		
	$\eta_{bj et1}$	•	•		
	n_{jet}	•	•		
	$n_{bj et}$	•	•		
	\cancel{E}_T	•	•		•
	H_T	•	•		•
high-level	m_T	•		•	•
	m_{T2}^W	•		•	•
	$\Delta\phi(W, l)$	•		•	
	$m(l, b)$	•		•	
	Centrality	•		•	
	Y	•		•	
	H_T -ratio	•		•	
	$\Delta r_{\min}(l, b)$	•		•	
	$\Delta\phi_{\min}(j_{1,2}, \cancel{E}_T)$	•		•	

PRIMERAS IMPLEMENTACIONES DE DEEP LEARNING

(Baldi et al. 2014, Guest et al. 2019, Sahin et al. 2016)

- ▶ Uso de redes profundas y poco profundas para discriminación de señales



DEEP LEARNING

GENERATIVE ADVERSARIAL NETWORKS

GENERATIVE ADVERSARIAL NETWORKS

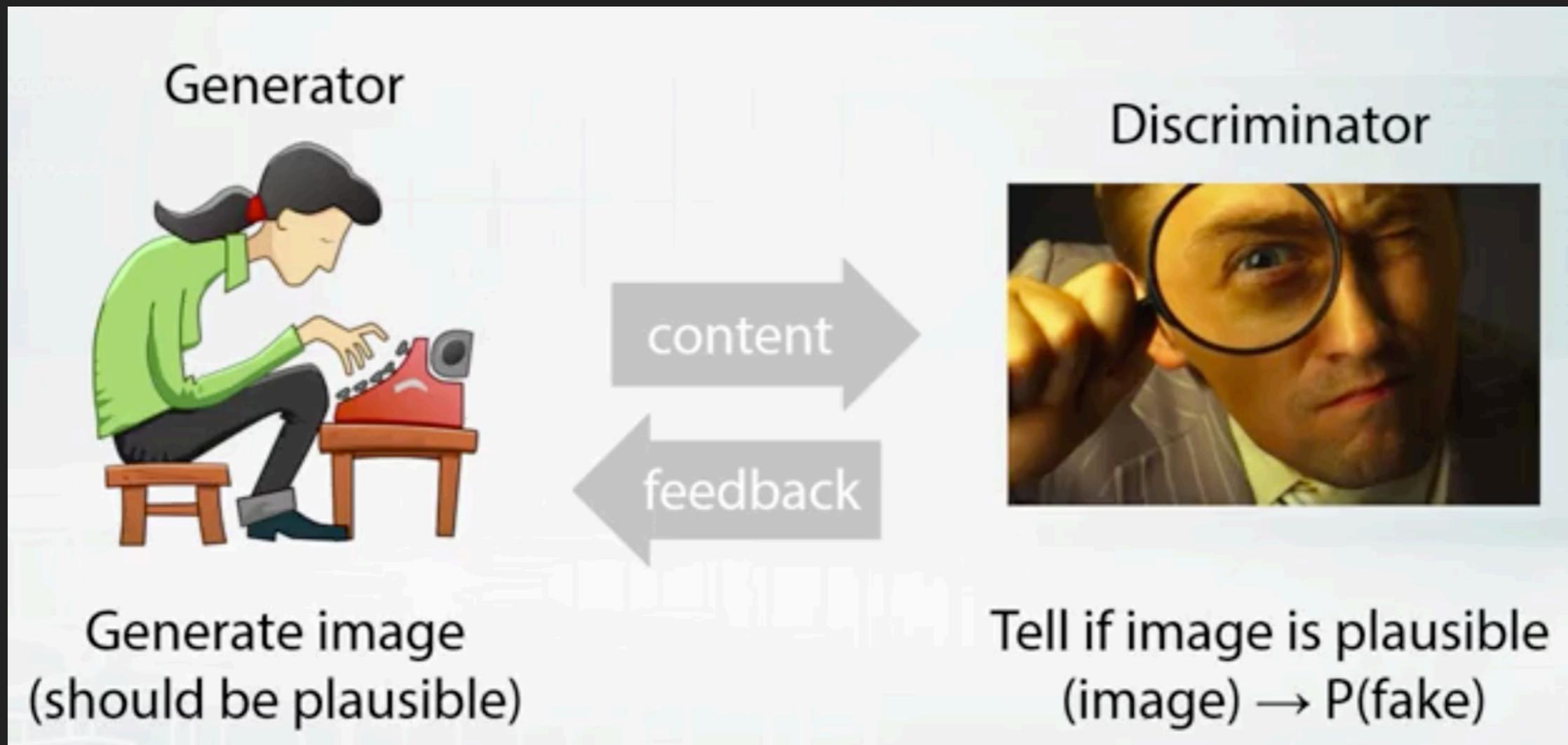
- ▶ Modelo generativo: Reconstruir $P(\mathbf{x})$

GENERATIVE ADVERSARIAL NETWORKS

- ▶ Modelo generativo: Reconstruir $P(\mathbf{x})$
- ▶ Generador

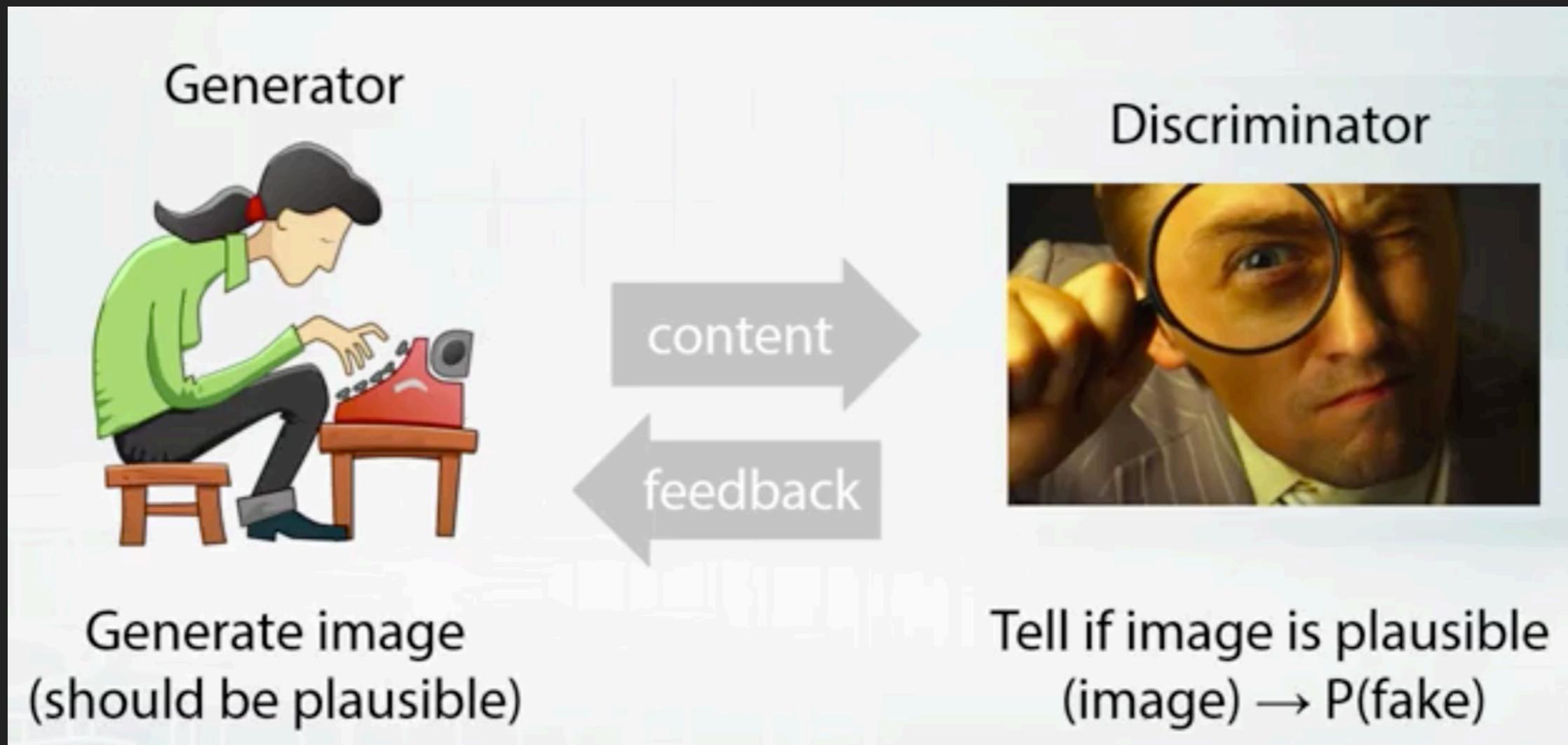
GENERATIVE ADVERSARIAL NETWORKS

- ▶ Modelo generativo: Reconstruir $P(\mathbf{x})$
- ▶ Generador



GENERATIVE ADVERSARIAL NETWORKS

- ▶ Modelo generativo: Reconstruir $P(\mathbf{x})$
- ▶ Generador
- ▶ Discriminador

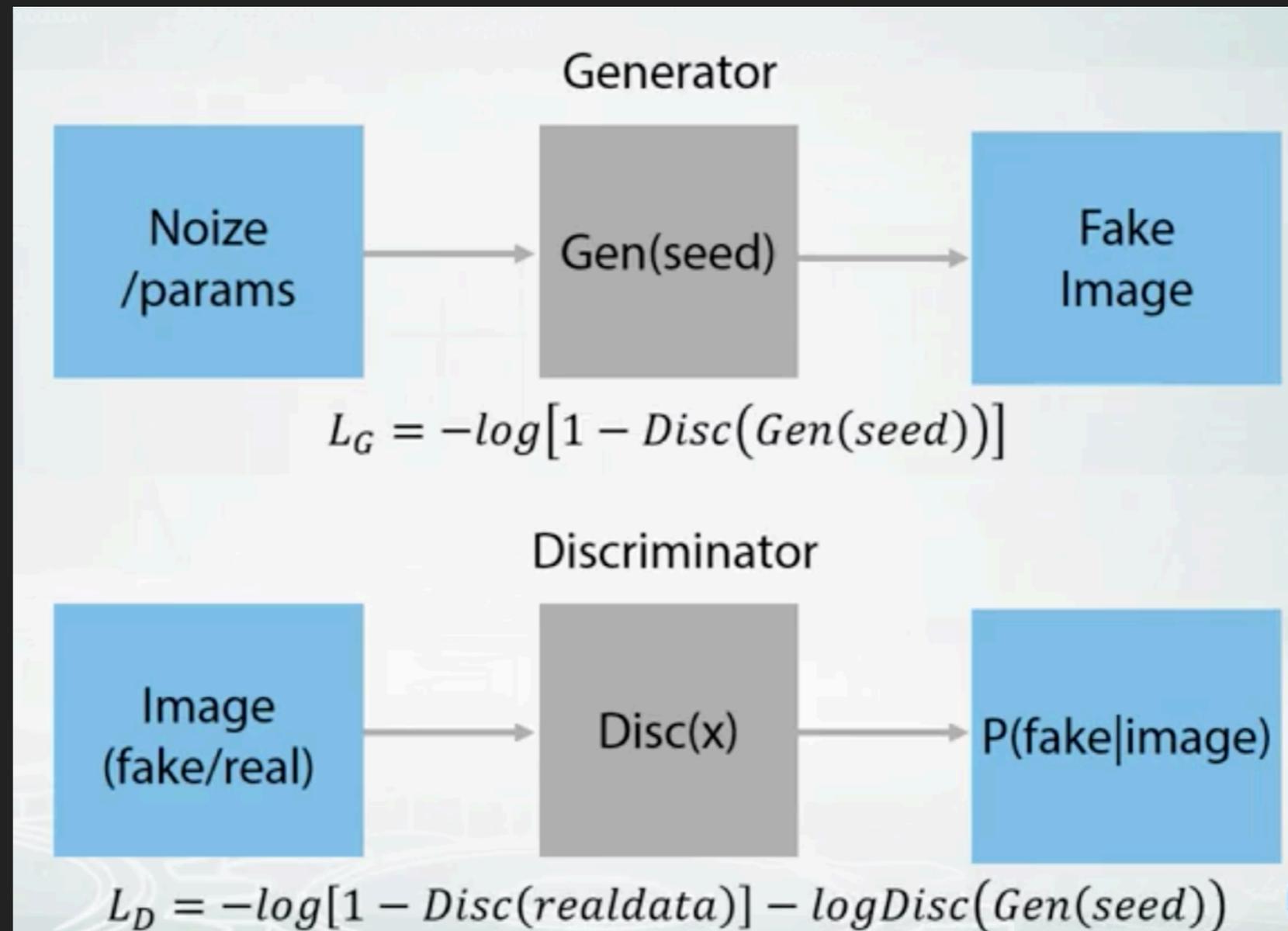


GENERATIVE ADVERSARIAL NETWORKS

- ▶ Modelo generativo: Reconstruir $P(\mathbf{x})$
- ▶ Generador
- ▶ Discriminador

GENERATIVE ADVERSARIAL NETWORKS

- ▶ Modelo generativo: Reconstruir $P(\mathbf{x})$
- ▶ Generador
- ▶ Discriminador



ADVERSARIAL DEBIASING

(Zhang et al. 2018)

ADVERSARIAL DEBIASING

(Zhang et al. 2018)

- ▶ Predicción: $\hat{Y} = F_p(X)$

ADVERSARIAL DEBIASING

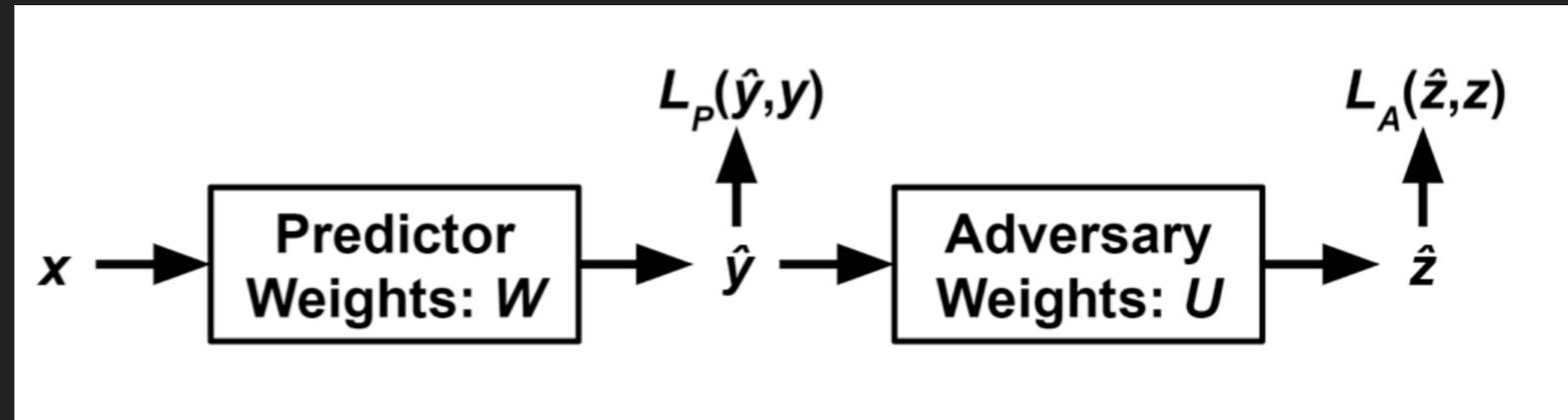
(Zhang et al. 2018)

- ▶ Predicción: $\hat{Y} = F_p(X)$
- ▶ Z: Variable protegida

ADVERSARIAL DEBIASING

- ▶ Predicción: $\hat{Y} = F_p(X)$
- ▶ Z: Variable protegida

(Zhang et al. 2018)



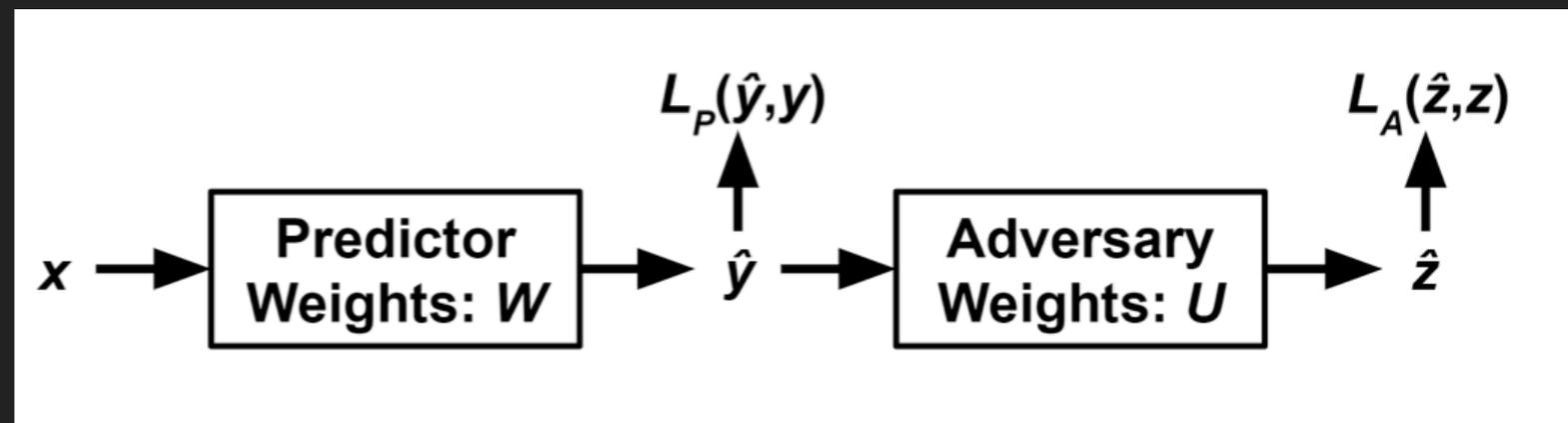
ADVERSARIAL DEBIASING

► Predicción: $\hat{Y} = F_p(X)$

► Z: Variable protegida

► Lo que queremos lograr es que

(Zhang et al. 2018)



ADVERSARIAL DEBIASING

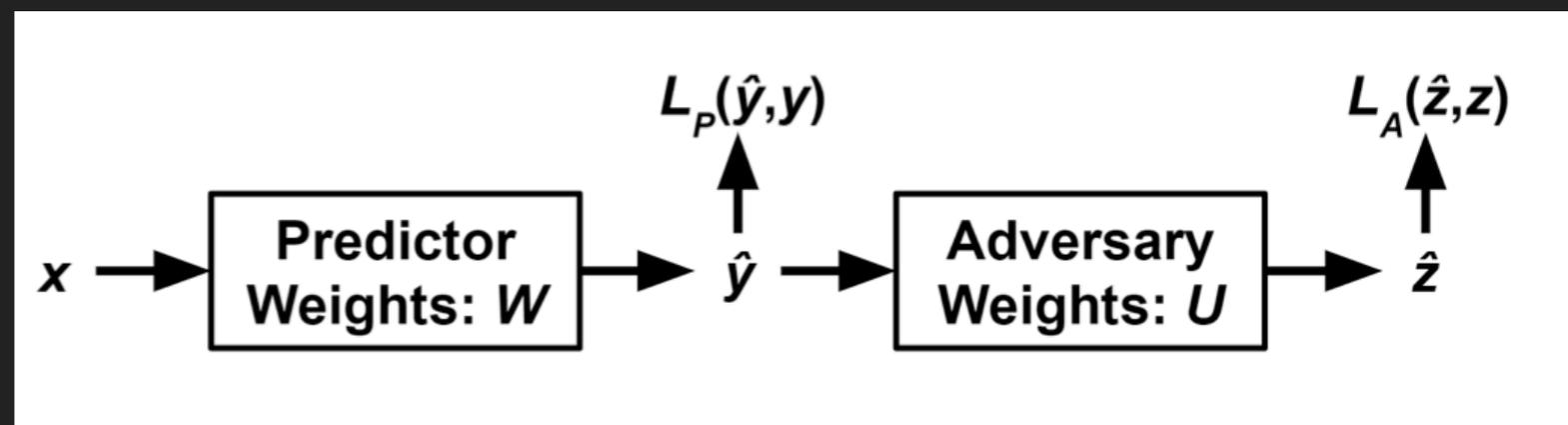
▶ Predicción: $\hat{Y} = F_p(X)$

▶ Z: Variable protegida

▶ Lo que queremos lograr es que

▶ Paridad demografica: $\hat{Y} \perp Z$

(Zhang et al. 2018)



ADVERSARIAL DEBIASING

▶ Predicción: $\hat{Y} = F_p(X)$

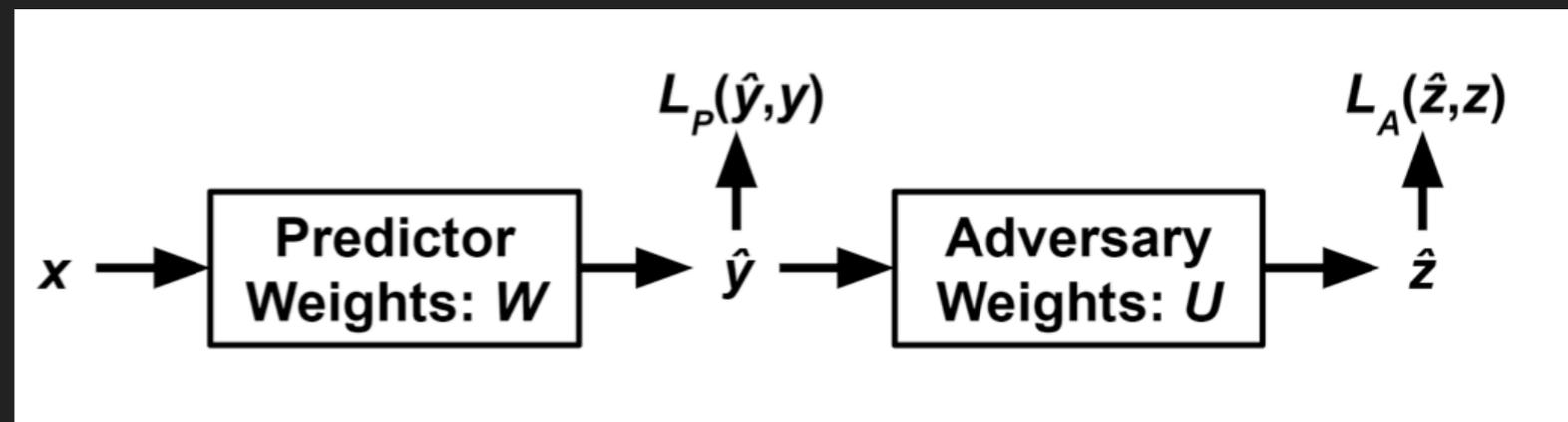
▶ Z: Variable protegida

▶ Lo que queremos lograr es que

▶ Paridad demografica: $\hat{Y} \perp Z$

▶ Equidad de oportunidad: $\hat{Y} \perp Z | Y$

(Zhang et al. 2018)



ADVERSARIAL DEBIASING

(Zhang et al. 2018)

▶ Predicción: $\hat{Y} = F_p(X)$

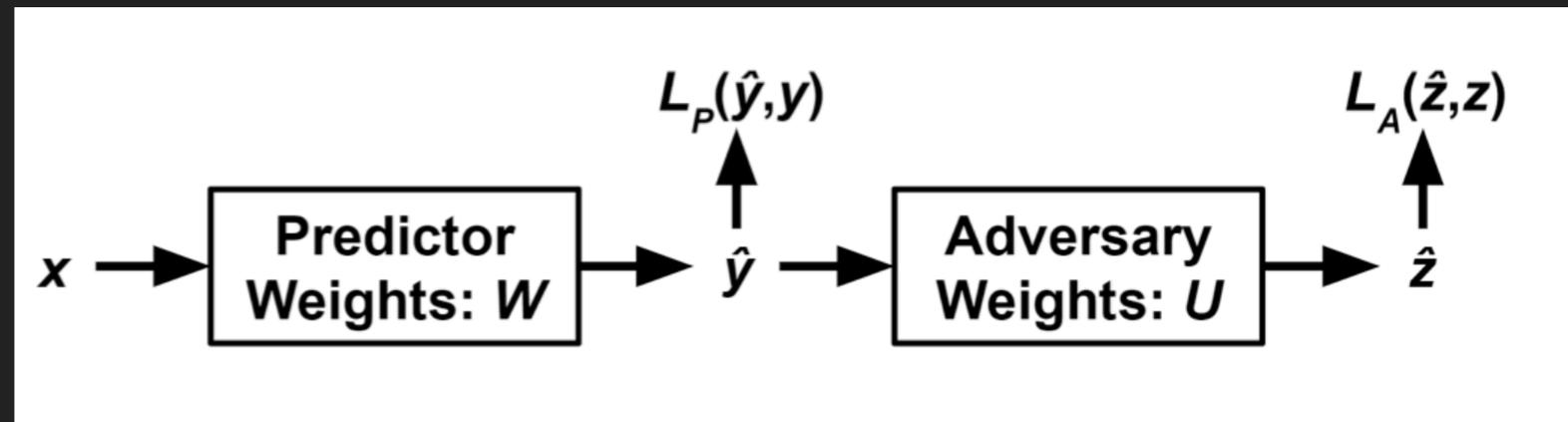
▶ Z: Variable protegida

▶ Lo que queremos lograr es que

▶ Paridad demografica: $\hat{Y} \perp Z$

▶ Equidad de oportunidad: $\hat{Y} \perp Z | Y$

▶ Equidad de probabilidades: $\hat{Y} \perp Z | Y = y$



ADVERSARIAL DEBIASING

▶ Predicción: $\hat{Y} = F_p(X)$

▶ Z: Variable protegida

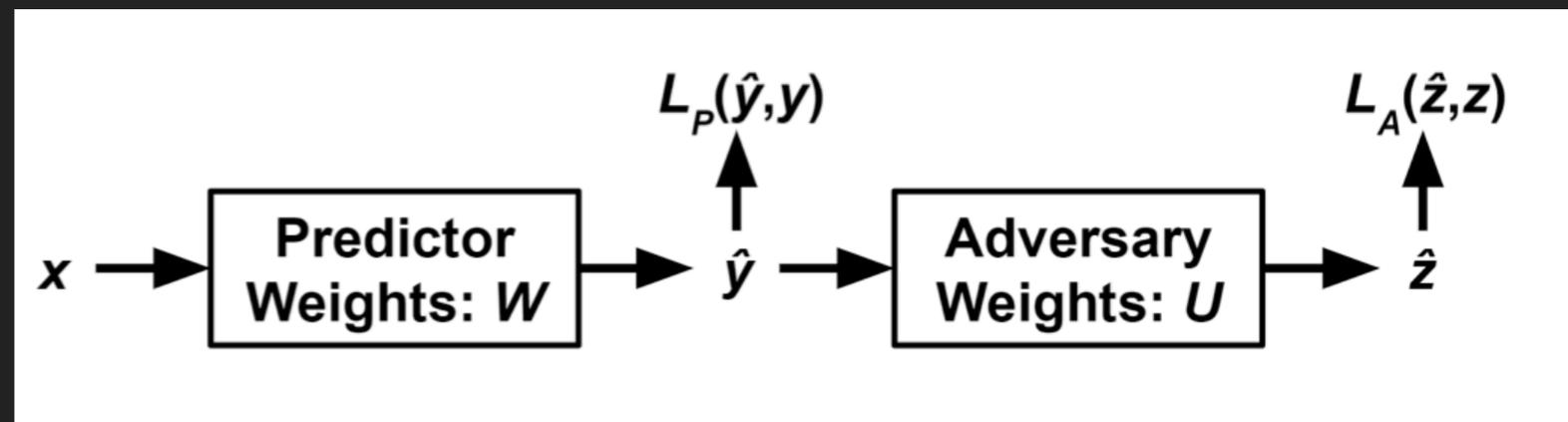
▶ Lo que queremos lograr es que

▶ Paridad demografica: $\hat{Y} \perp Z$

▶ Equidad de oportunidad: $\hat{Y} \perp Z | Y$

▶ Equidad de probabilidades: $\hat{Y} \perp Z | Y = y$

(Zhang et al. 2018)



Actualizando param W con:

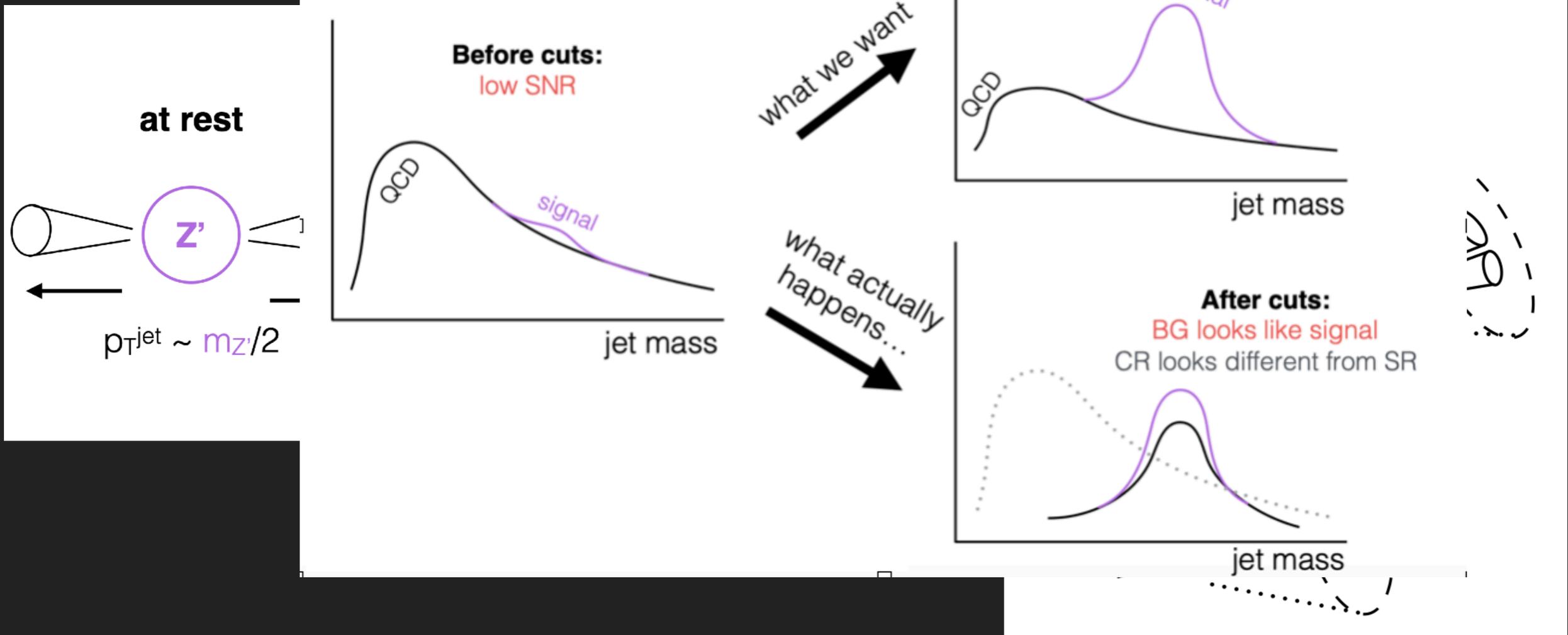
$$\nabla_W L_P - \text{proj}_{\nabla_W L_A} \nabla_W L_P - \alpha \nabla_W L_A$$

DECORRELATED JET SUBSTRUCTURE TAGGING USING ADVERSARIAL NEURAL NETWORKS

(Shimmin et al. 2017)

- ▶ Reducir correlaciones a la masa del jet

- ▶ Distorsión en el background

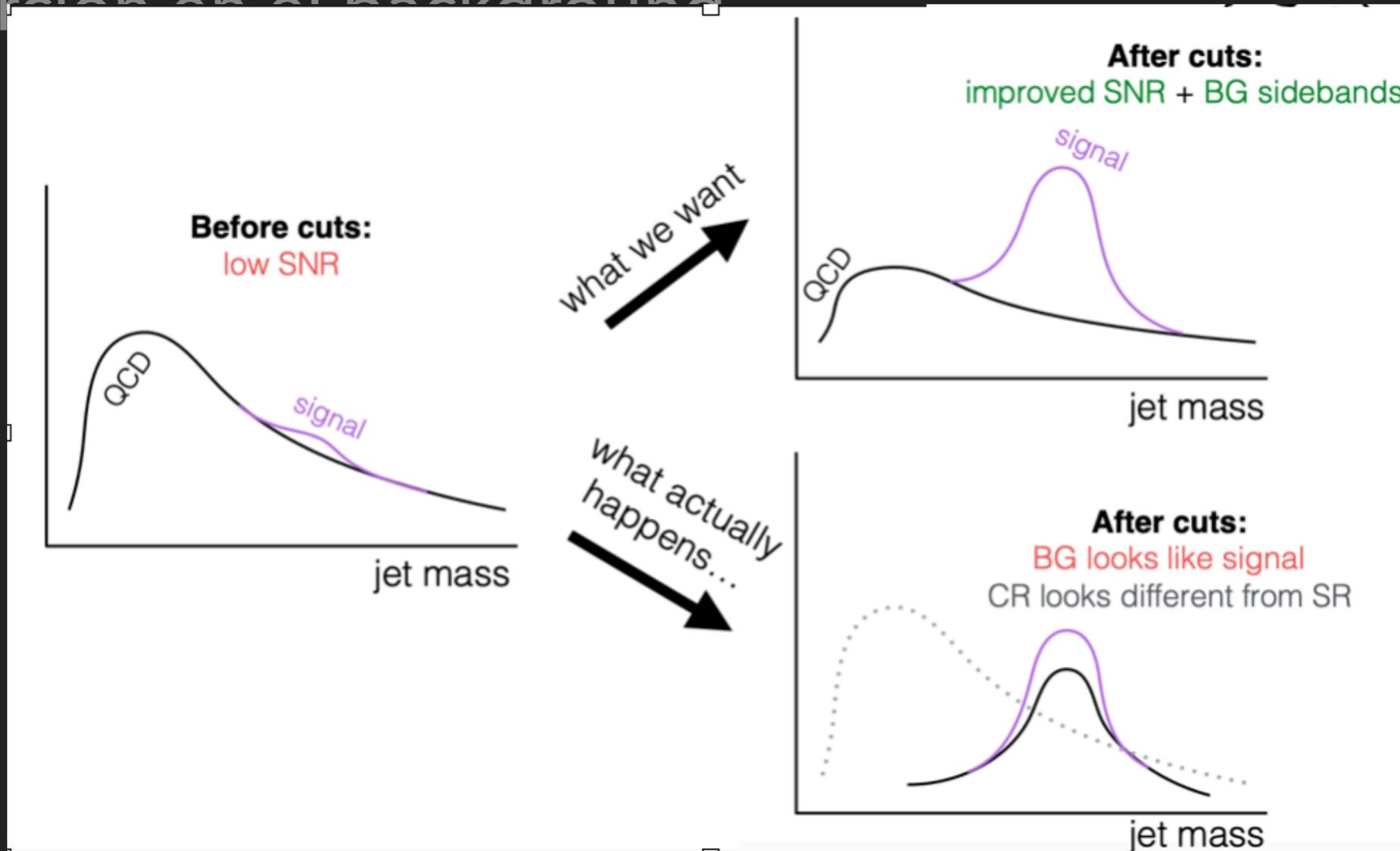


DECORRELATED JET SUBSTRUCTURE TAGGING USING ADVERSARIAL NEURAL NETWORKS

(Shimmin et al. 2017)

▶ Reducir correlaciones a la masa del jet

▶ Distorsión en el background



DECORRELATED JET SUBSTRUCTURE TAGGING USING ADVERSARIAL NEURAL NETWORKS

(Shimmin et al. 2017)

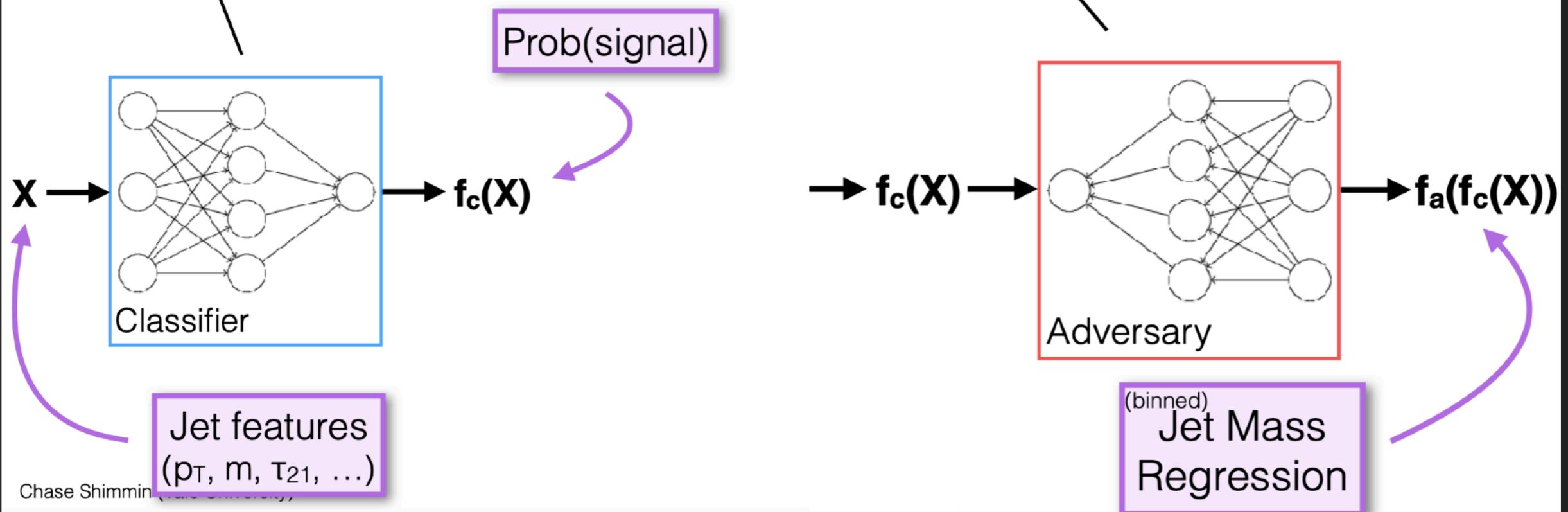
DECORRELATED JET SUBSTRUCTURE TAGGING USING ADVERSARIAL NEURAL NETWORKS

(Shimmin et al. 2017)

Basic idea:

Classifier is trained to identify signal jets

Adversary is trained to predict jet mass

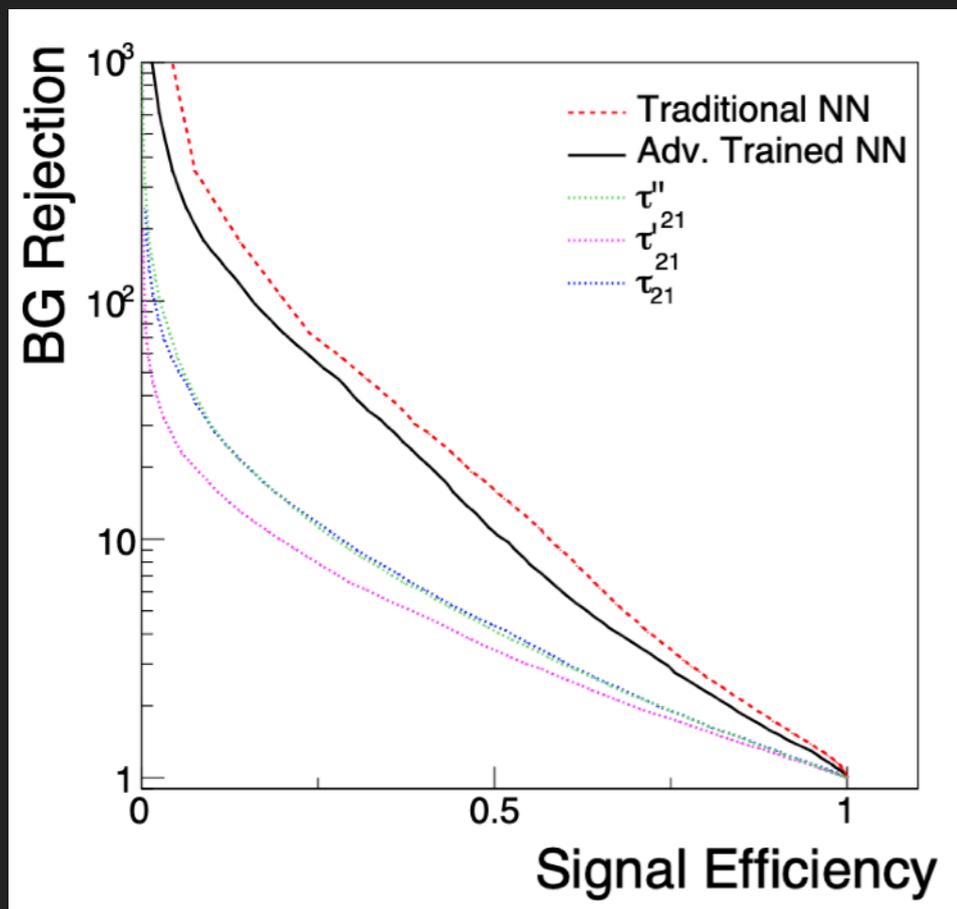


Chase Shimmin

DECORRELATED JET SUBSTRUCTURE TAGGING USING ADVERSARIAL NEURAL NETWORKS

(Shimmin et al. 2017)

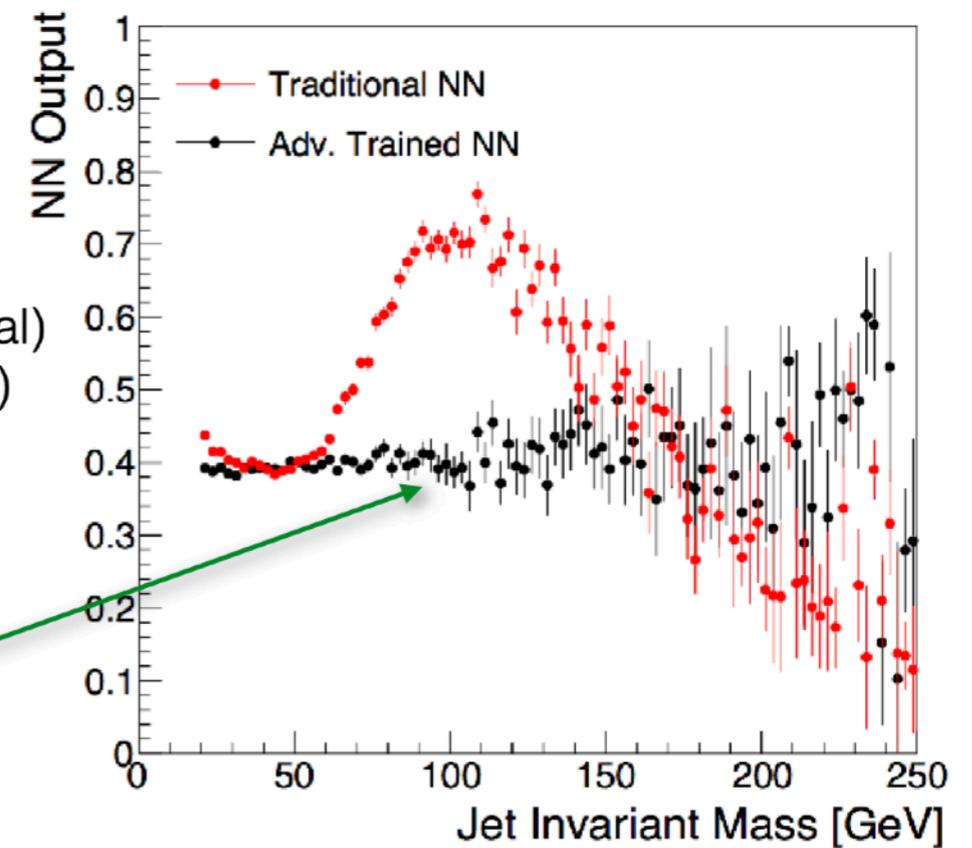
► De-correlación de la masa del jet



Training on ~200k
MC events:

Sherpa γ +jet (BG)
MG5 γ +Z' (Signal)
Pythia + Delphes (Both)

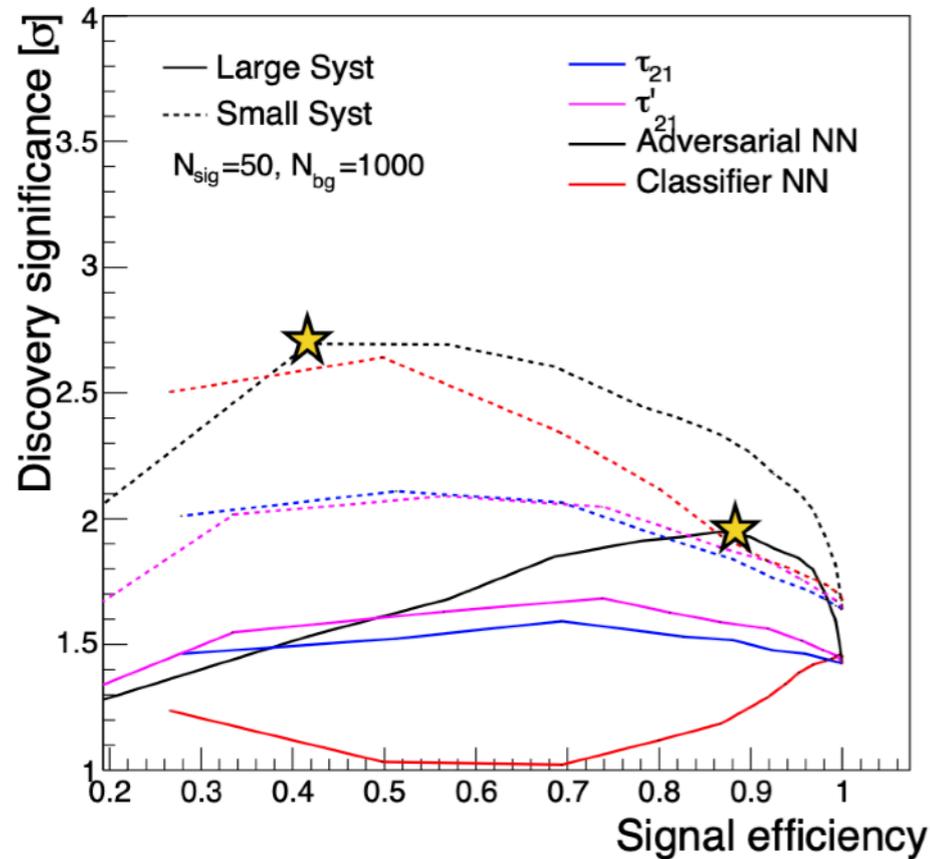
✓ Tagger profile
much flatter



DECORRELATED JET SUBSTRUCTURE TAGGING USING ADVERSARIAL NEURAL NETWORKS

(Shimmin et al. 2017)

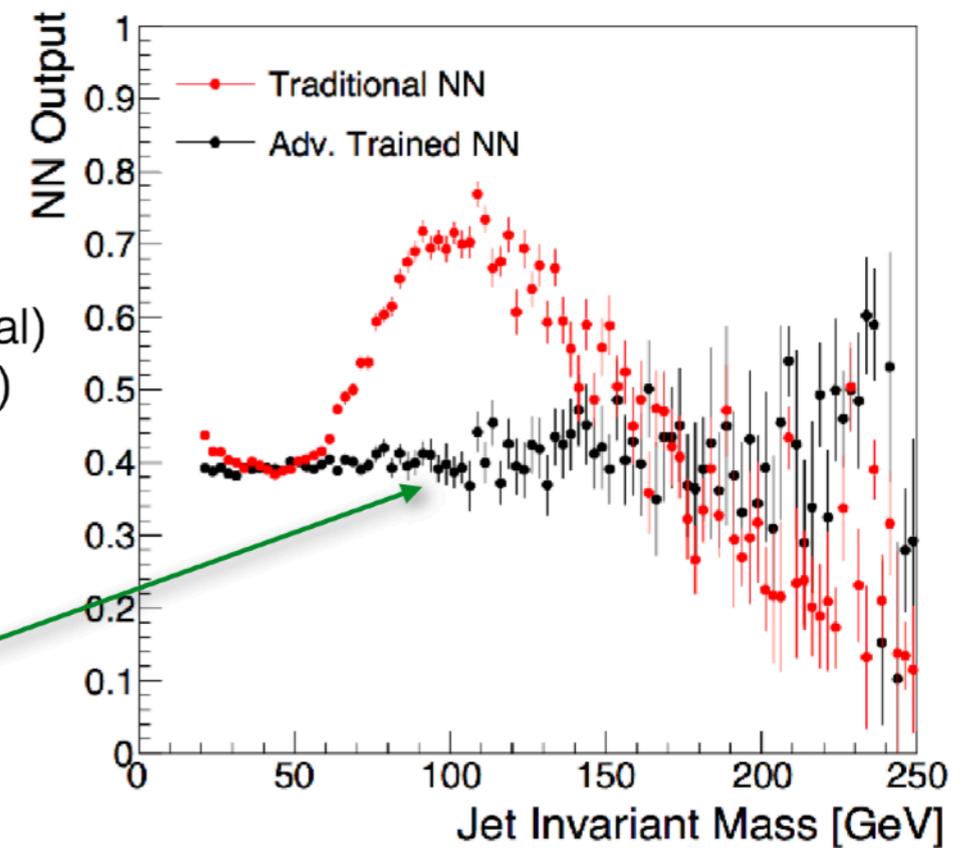
► De-correlación de la masa del jet



Training on ~200k
MC events:

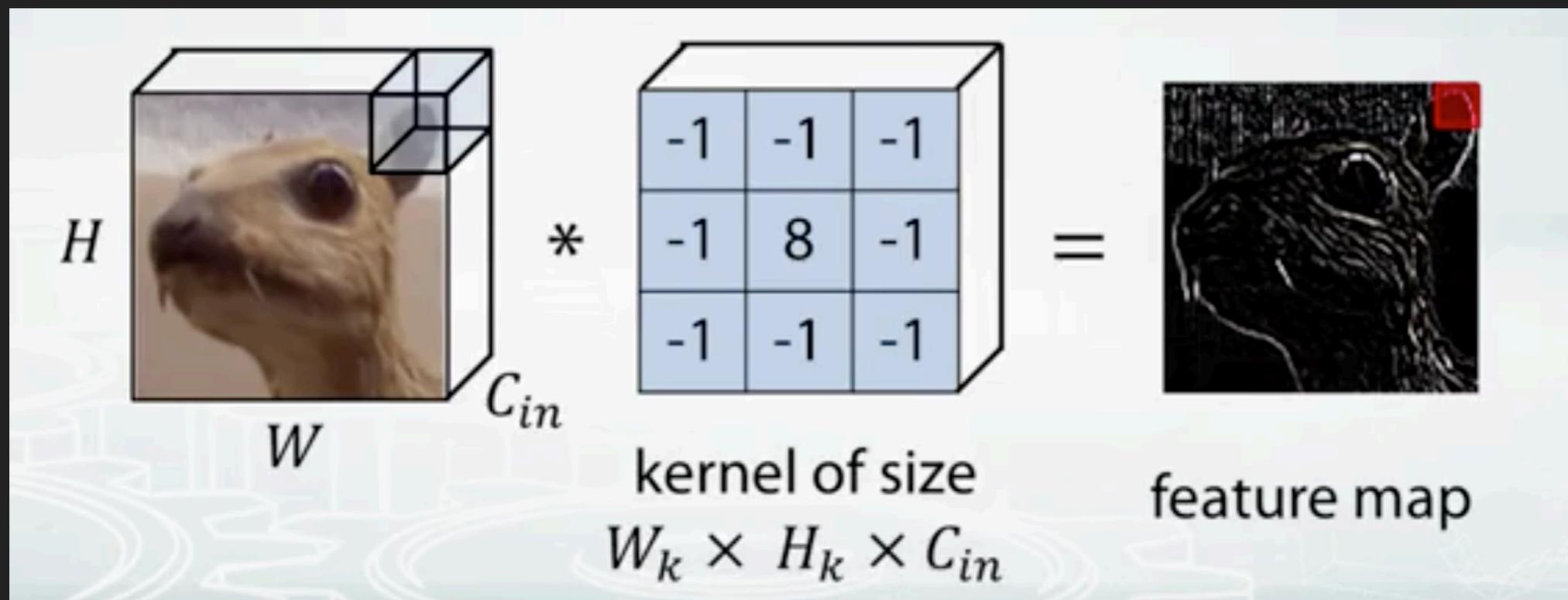
Sherpa γ +jet (BG)
 MG5 γ +Z' (Signal)
 Pythia + Delphes (Both)

✓ Tagger profile
much flatter



REDES NEURONALES CONVOLUCIONALES (CNN) [1]

- ▶ W es el ancho
- ▶ H el alto
- ▶ C_{in} es la profundidad = (3=RGB para una imagen)

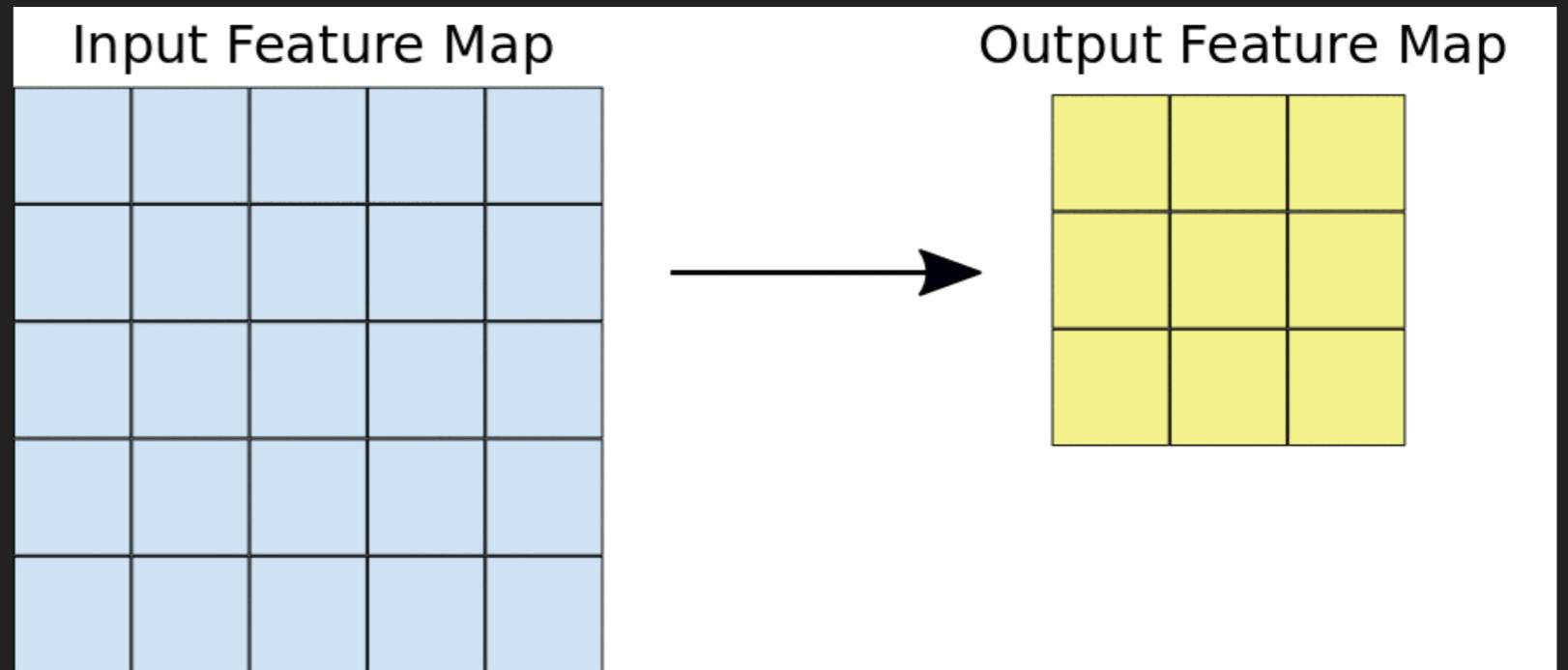


REDES NEURONALES CONVOLUCIONALES (CNN) [2]

- ▶ Kernel filters y Max pooling
- ▶ Parametros son compartidos en la red

REDES NEURONALES CONVOLUCIONALES (CNN) [2]

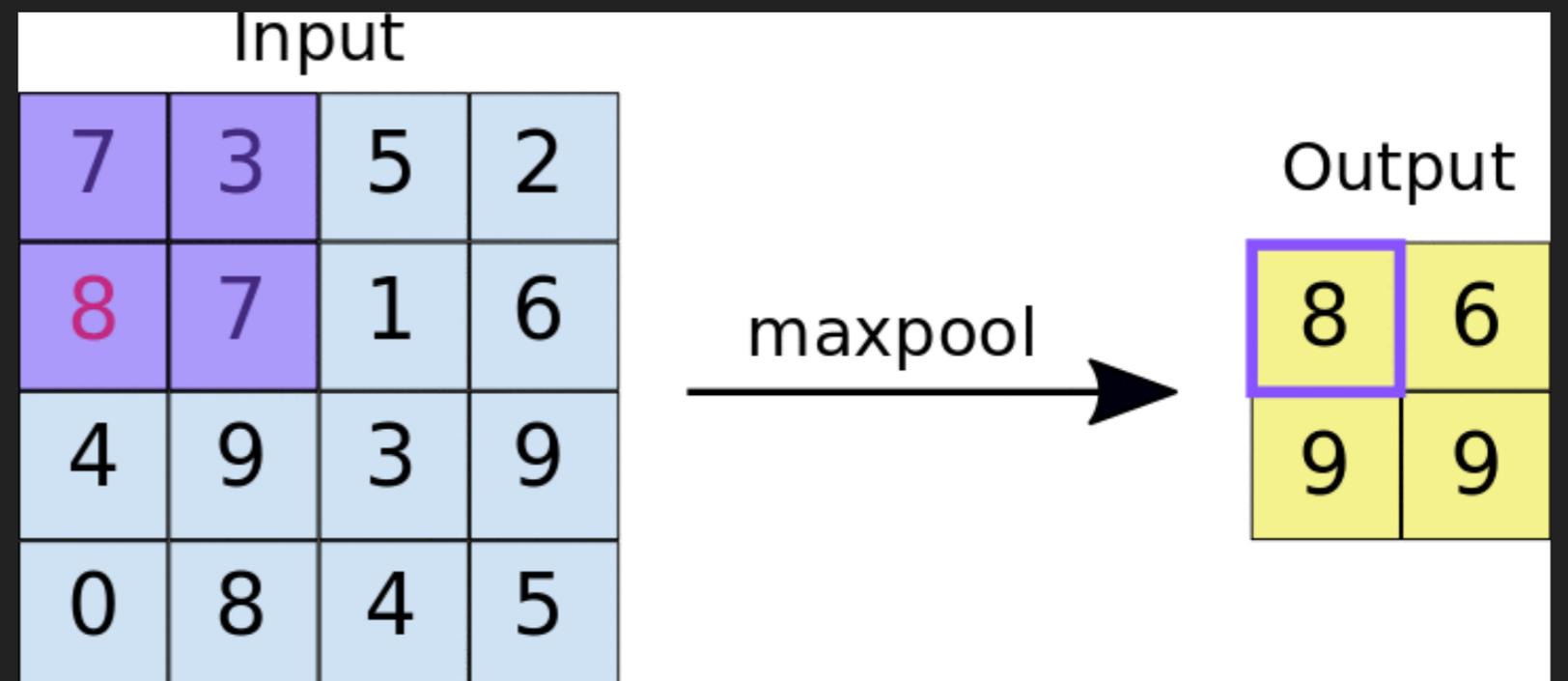
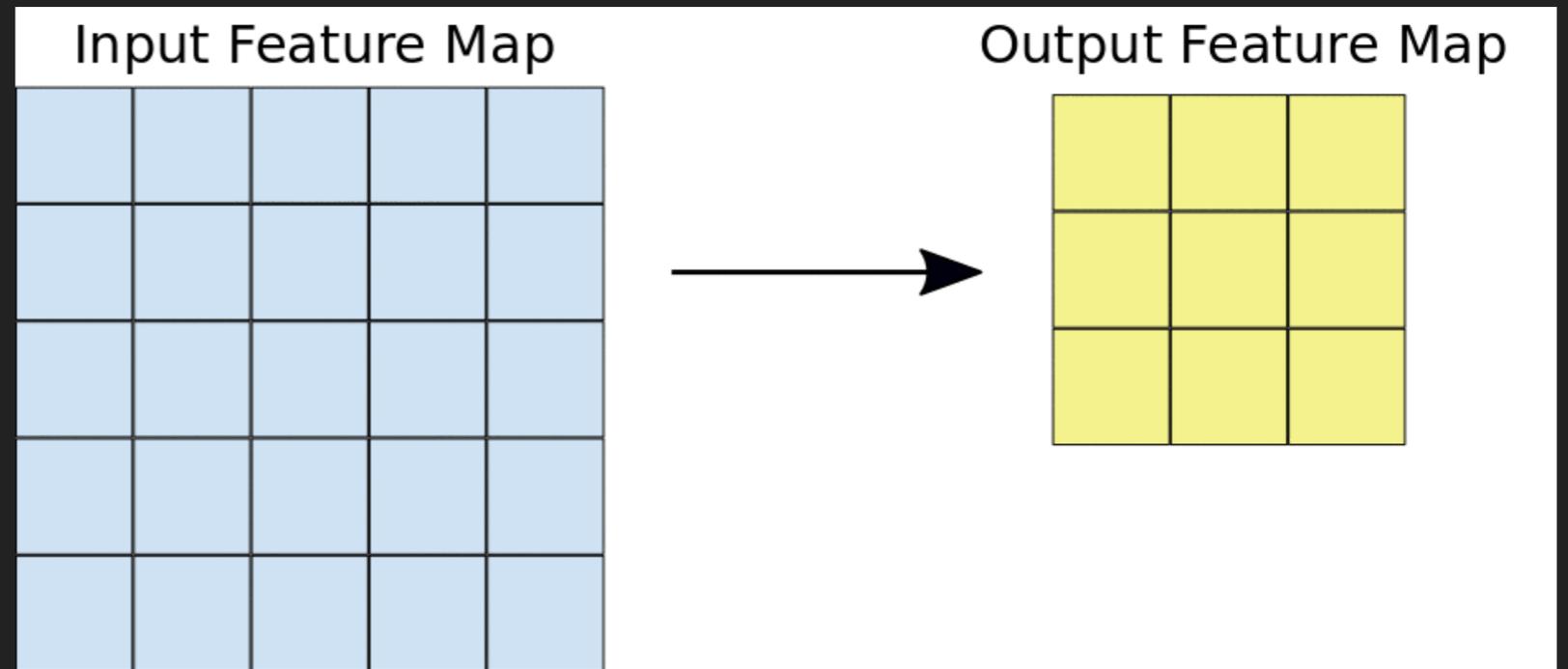
- ▶ Kernel filters y Max pooling
- ▶ Parametros son compartidos en la red



REDES NEURONALES CONVOLUCIONALES (CNN) [2]

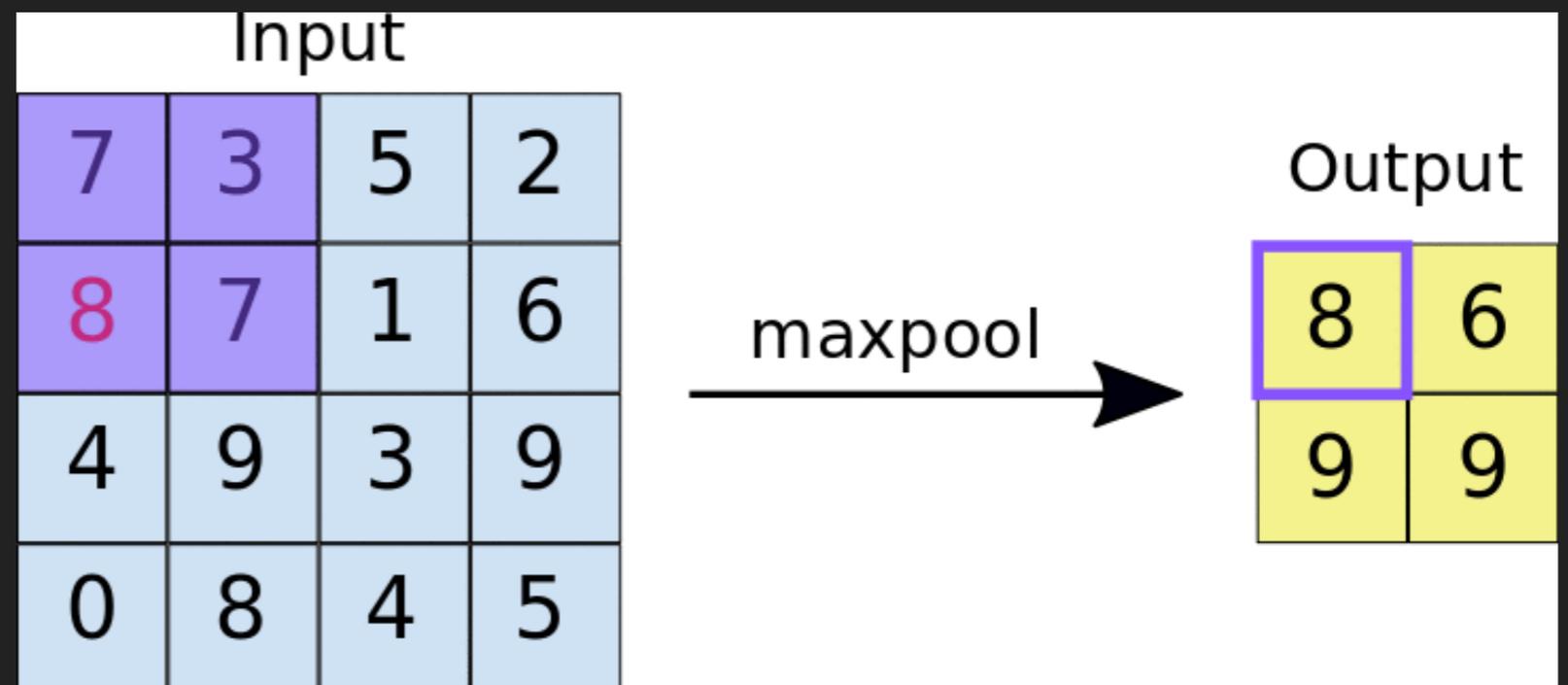
- ▶ Kernel filters y Max pooling

- ▶ Parametros son compartidos en la red



REDES NEURONALES CONVOLUCIONALES (CNN) [2]

- ▶ Kernel filters y Max pooling
- ▶ Parametros son compartidos en la red



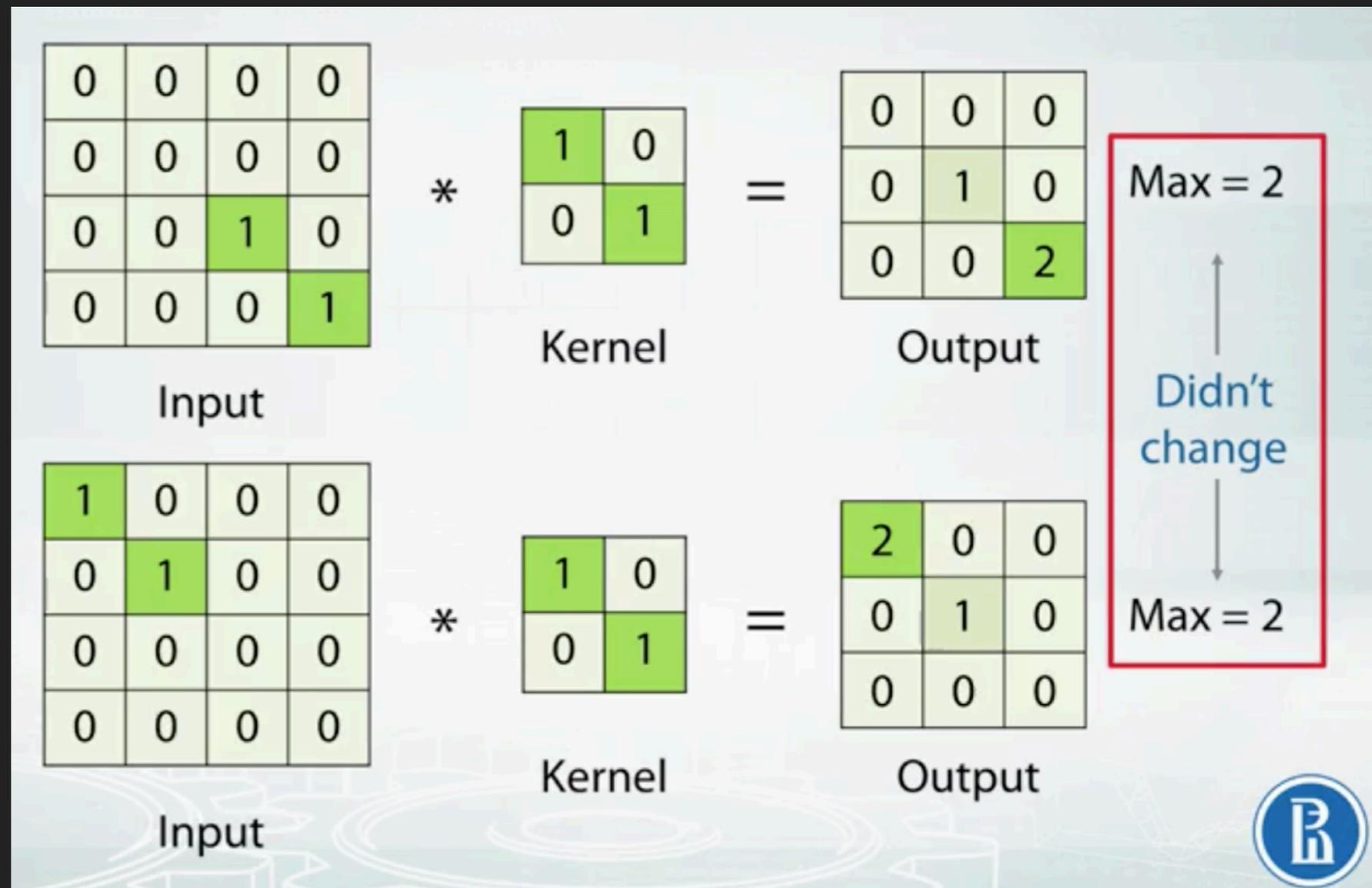
REDES NEURONALES CONVOLUCIONALES (CNN) [2]

- ▶ Kernel filters y Max pooling

- ▶ Parametros son compartidos en la red

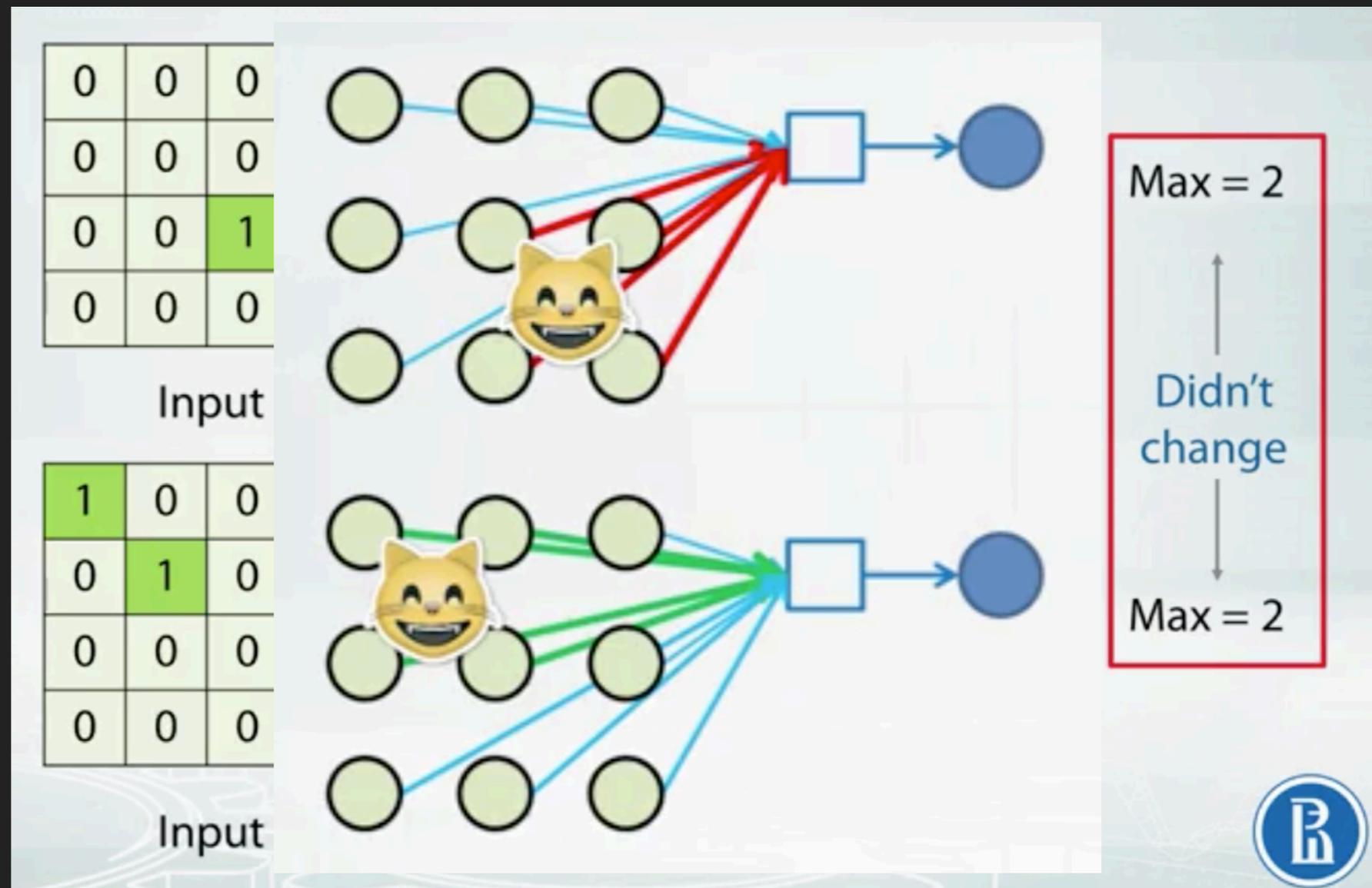
REDES NEURONALES CONVOLUCIONALES (CNN) [2]

- ▶ Kernel filters y Max pooling
- ▶ Parametros son compartidos en la red



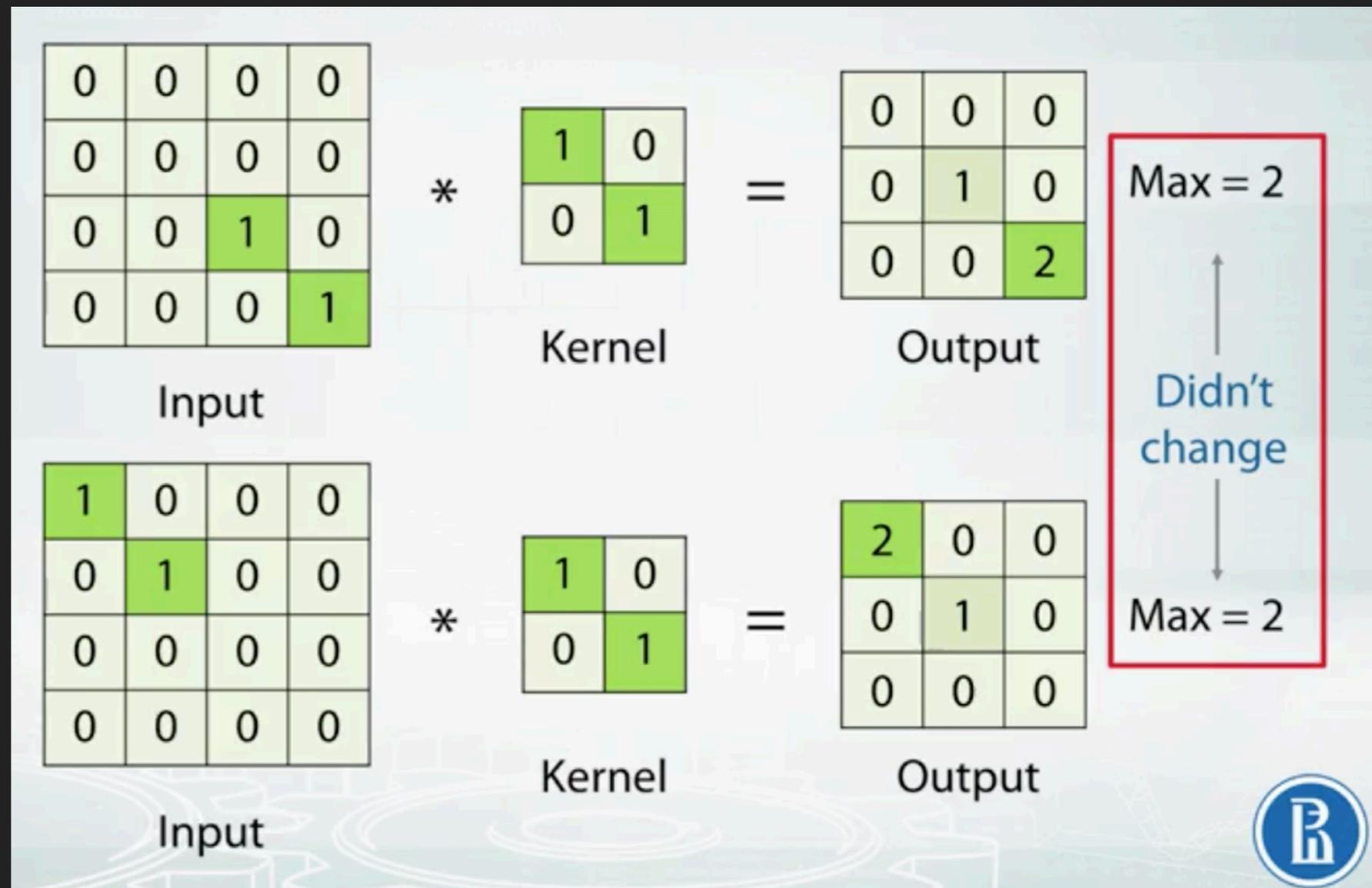
REDES NEURONALES CONVOLUCIONALES (CNN) [2]

- ▶ Kernel filters y Max pooling
- ▶ Parametros son compartidos en la red

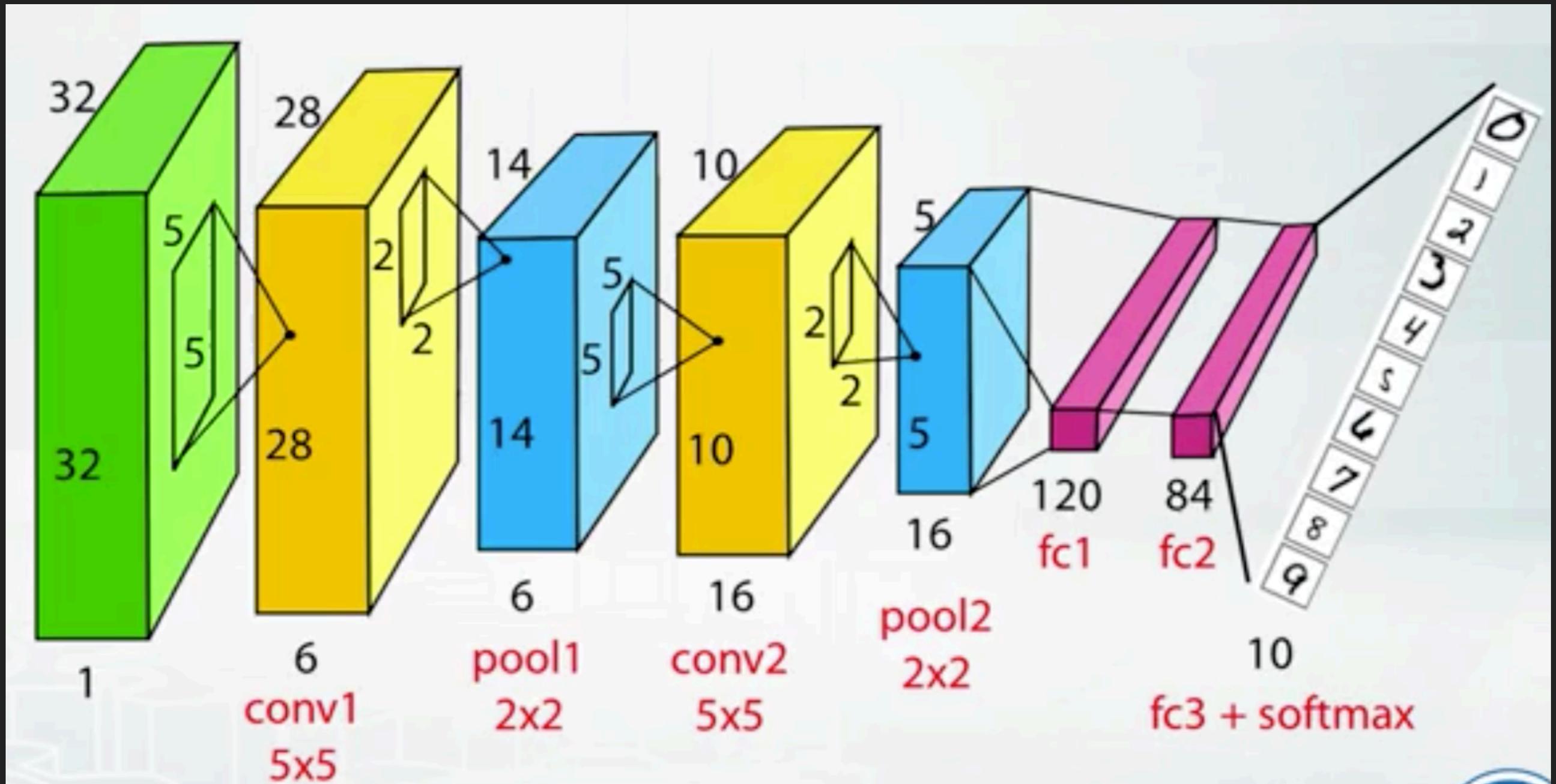


REDES NEURONALES CONVOLUCIONALES (CNN) [2]

- ▶ Kernel filters y Max pooling
- ▶ Parametros son compartidos en la red



REDES NEURONALES CONVOLUCIONALES (CNN) [3]

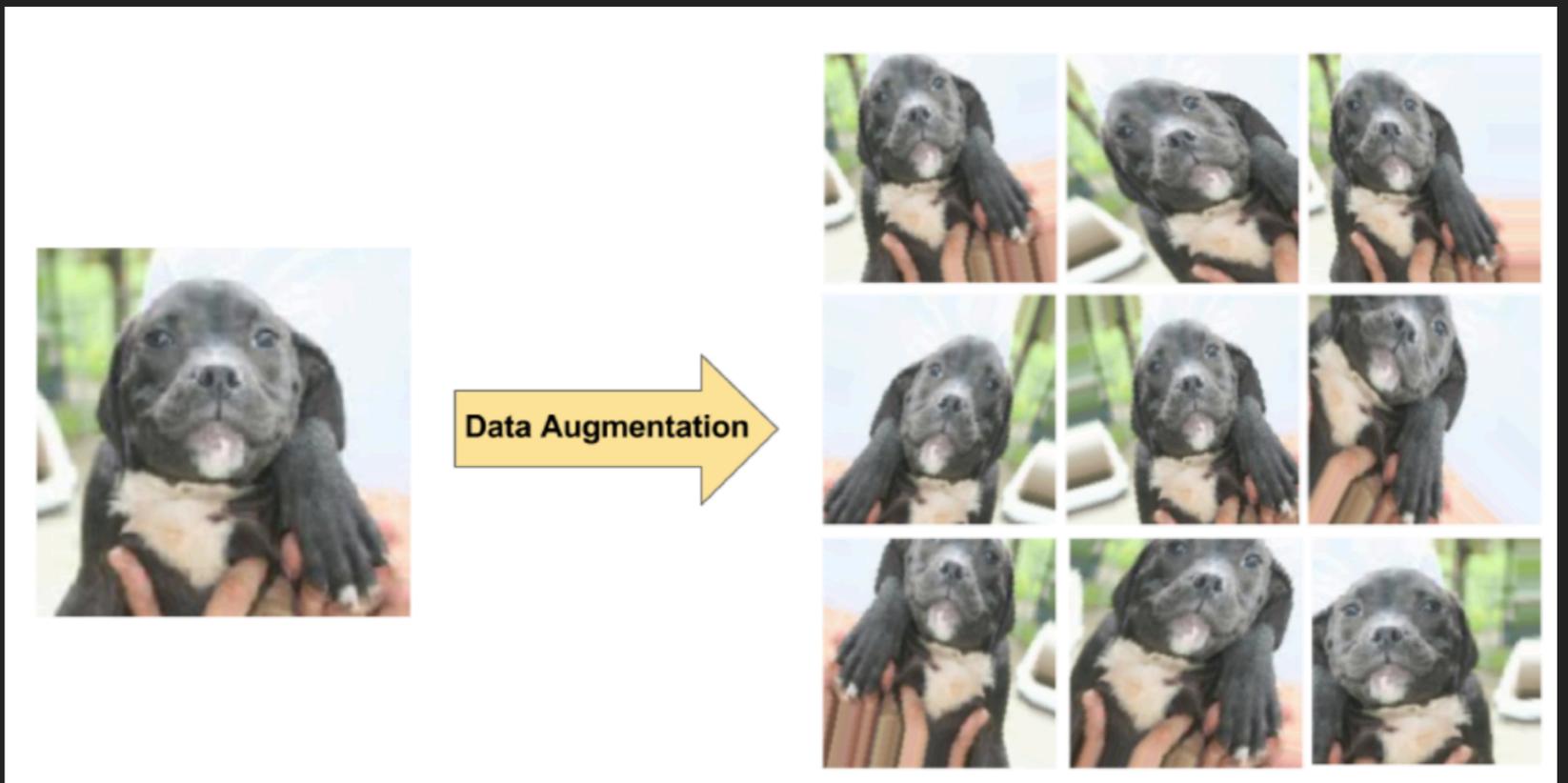


REDES NEURONALES CONVOLUCIONALES (CNN) [3]

- ▶ Aumento de datos
- ▶ Regularización por abandono

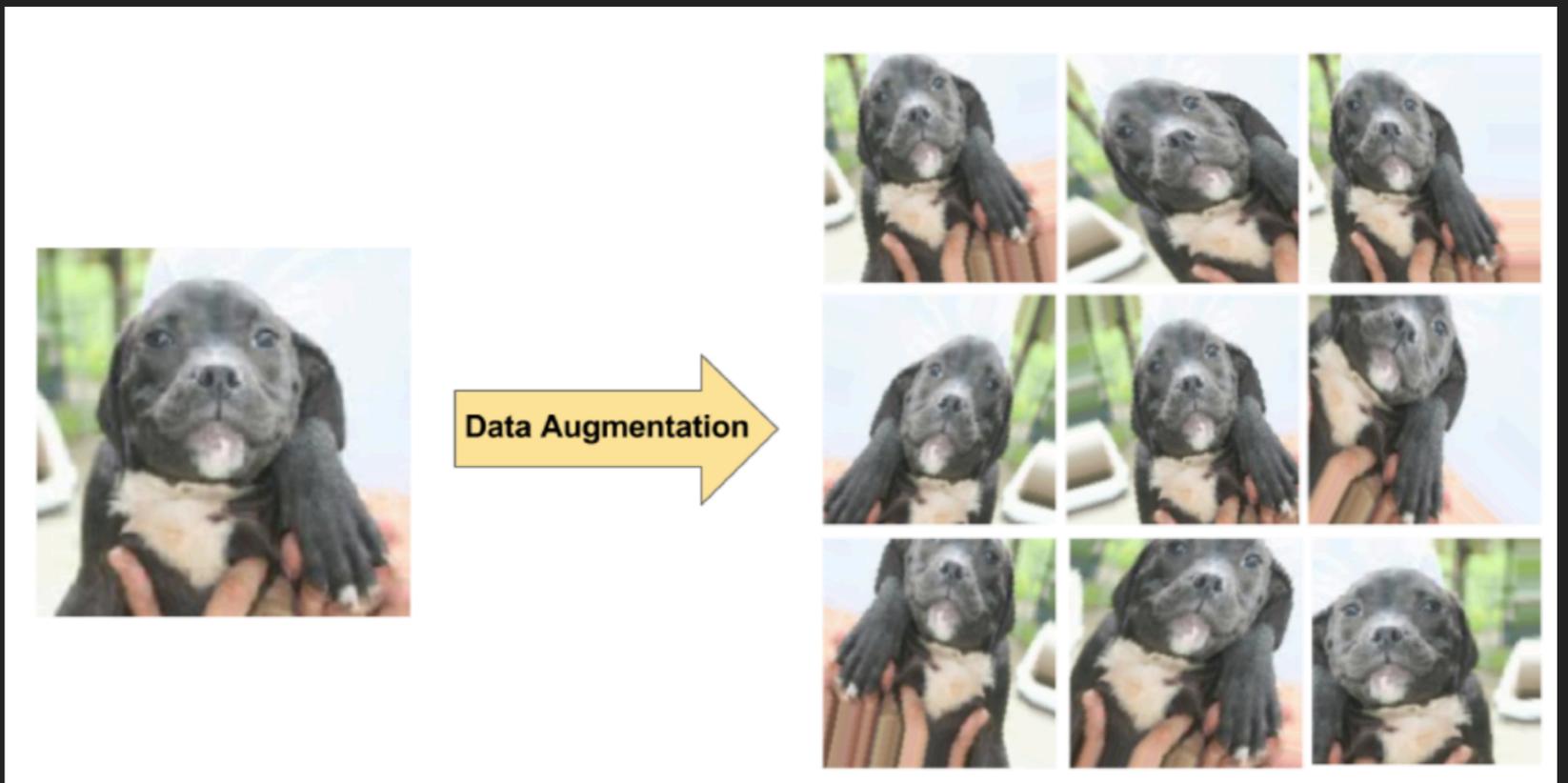
REDES NEURONALES CONVOLUCIONALES (CNN) [3]

- ▶ Aumento de datos
- ▶ Regularización por abandono



REDES NEURONALES CONVOLUCIONALES (CNN) [3]

- ▶ Aumento de datos
- ▶ Regularización por abandono



REDES NEURONALES CONVOLUCIONALES (CNN) [3]

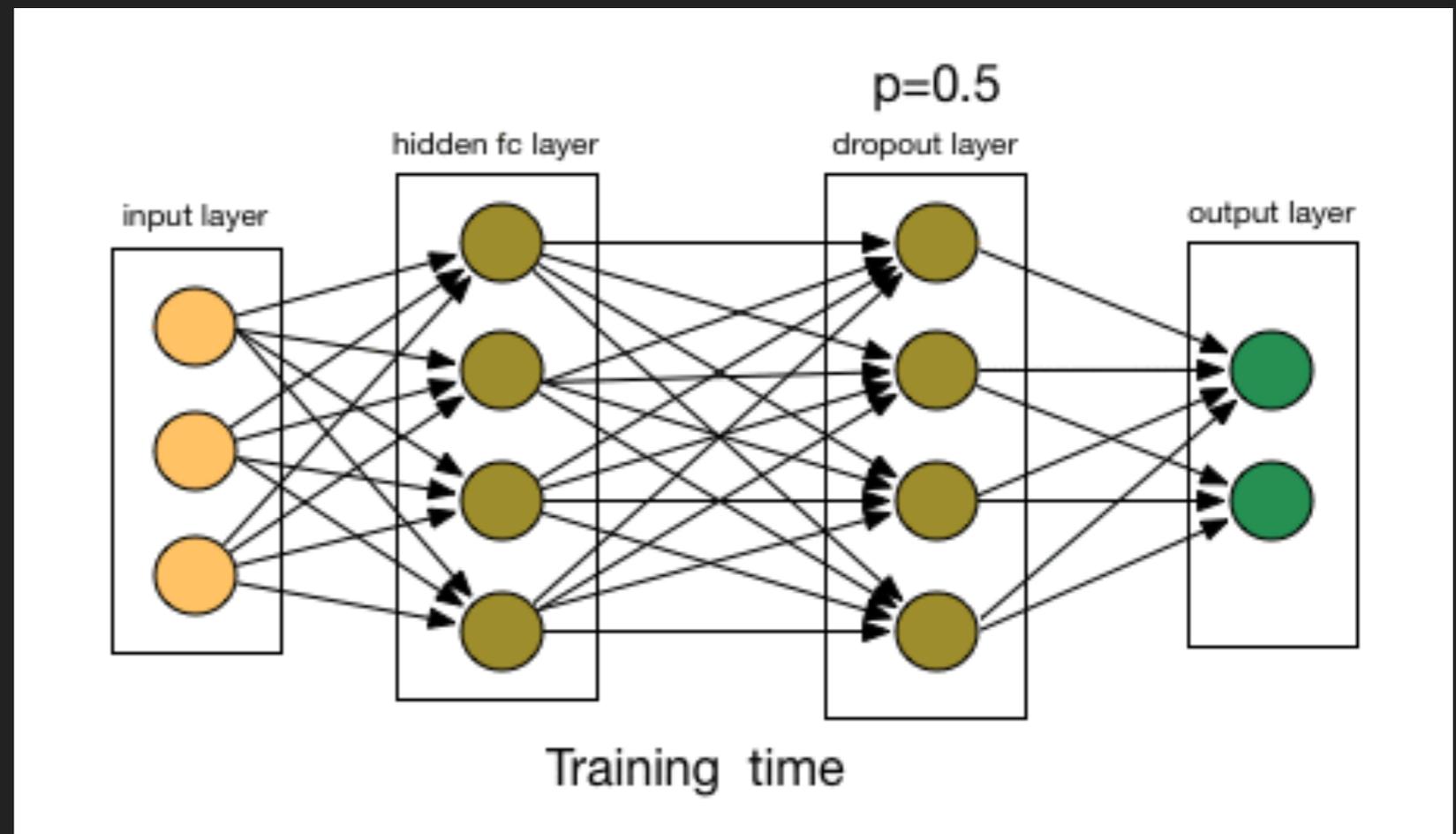
- ▶ Aumento de datos
- ▶ Regularización por abandono

REDES NEURONALES CONVOLUCIONALES (CNN) [3]

- ▶ Aumento de datos
- ▶ Regularización por abandono

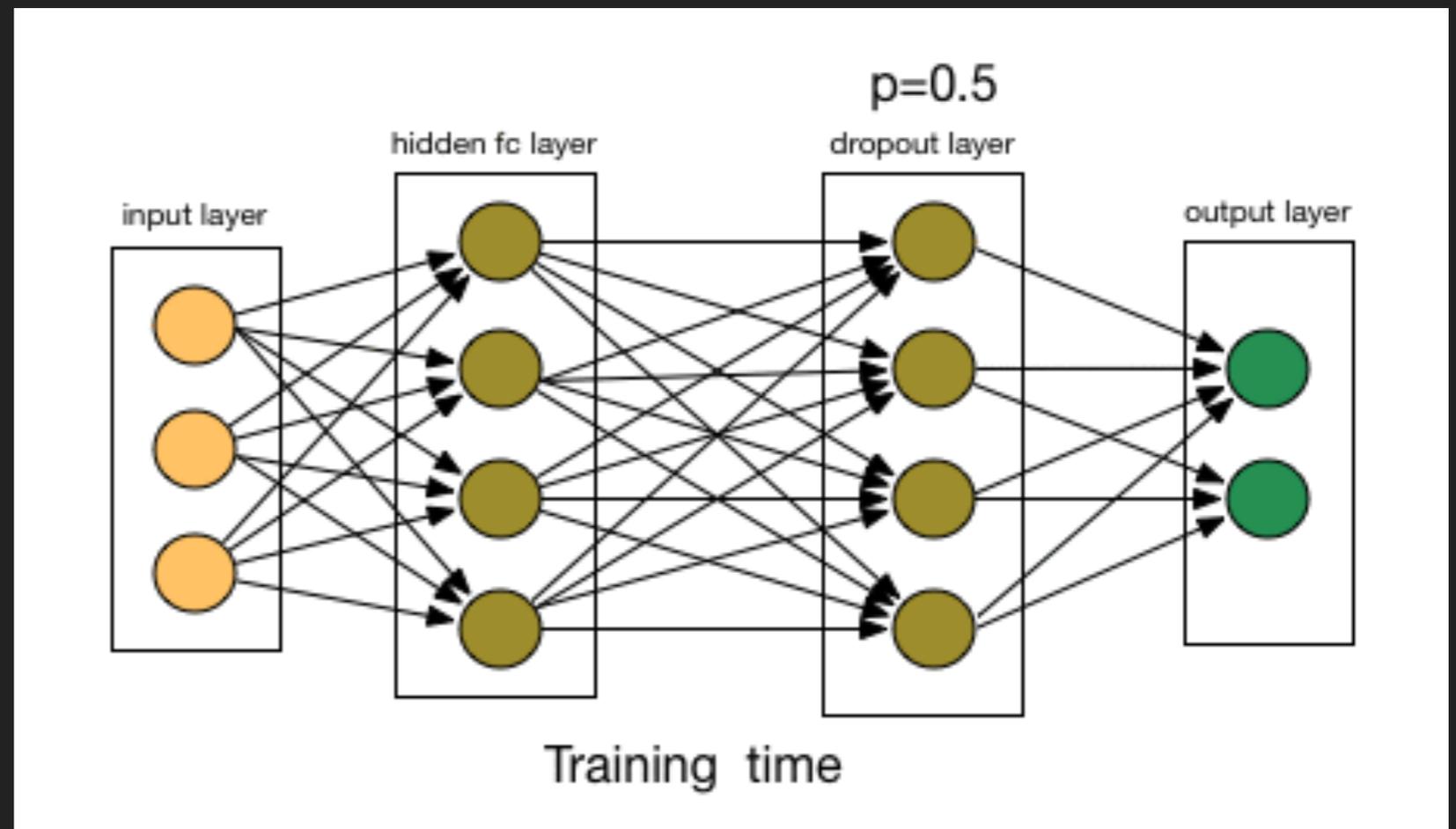
REDES NEURONALES CONVOLUCIONALES (CNN) [3]

- ▶ Aumento de datos
- ▶ Regularización por abandono



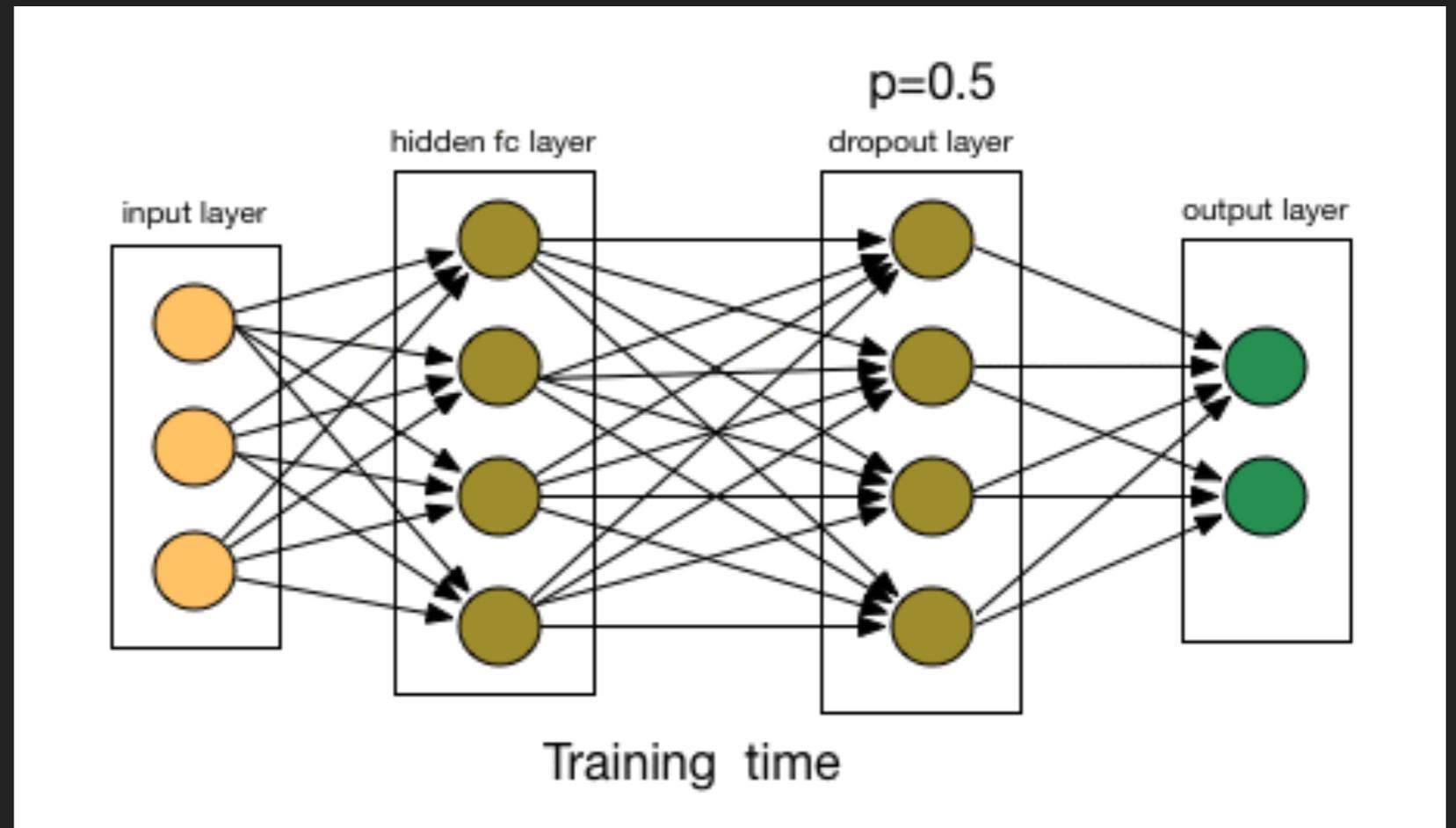
REDES NEURONALES CONVOLUCIONALES (CNN) [3]

- ▶ Aumento de datos
- ▶ Regularización por abandono



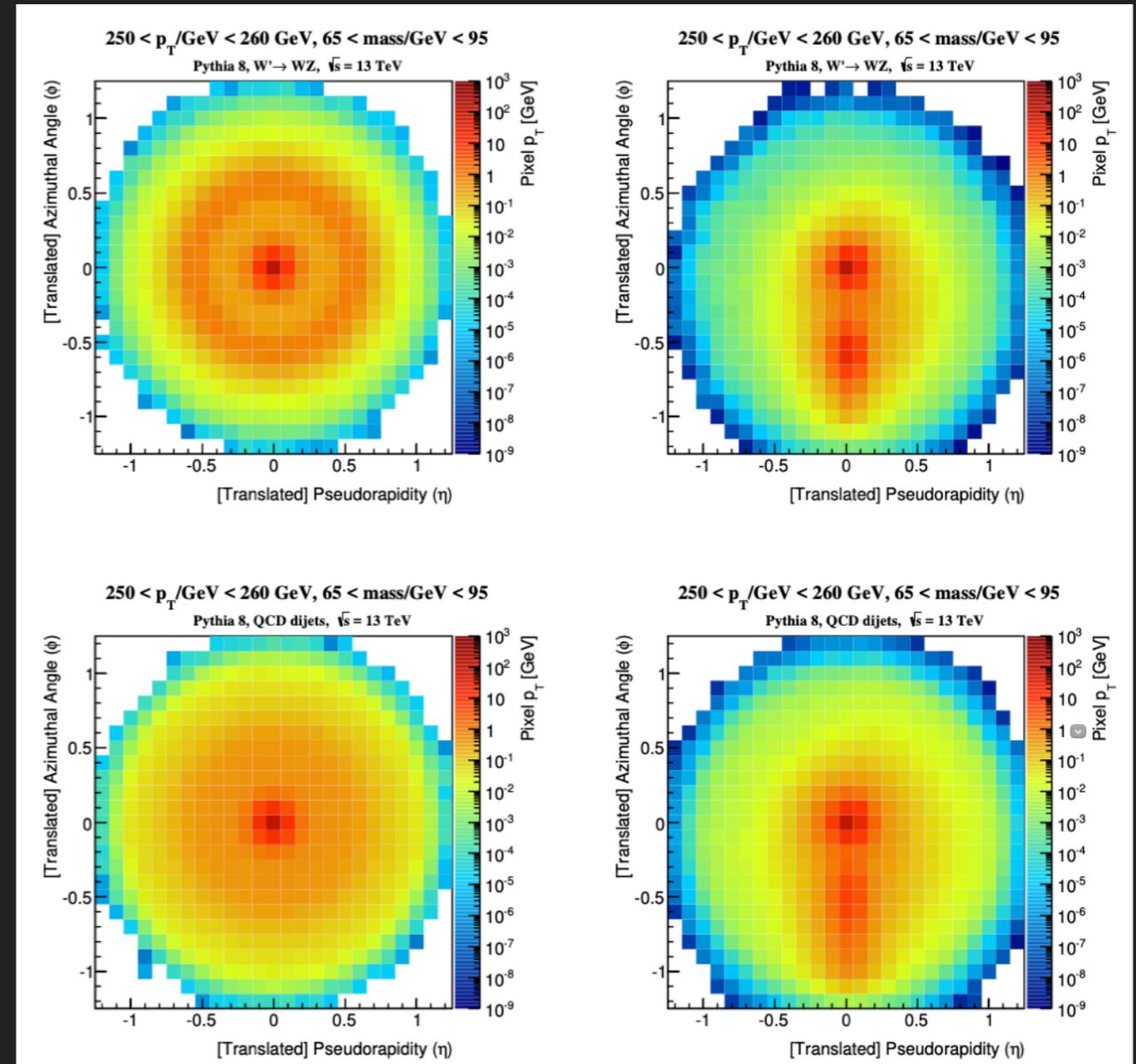
REDES NEURONALES CONVOLUCIONALES (CNN) [3]

- ▶ Aumento de datos
- ▶ Regularización por abandono



CNN EN EL LHC

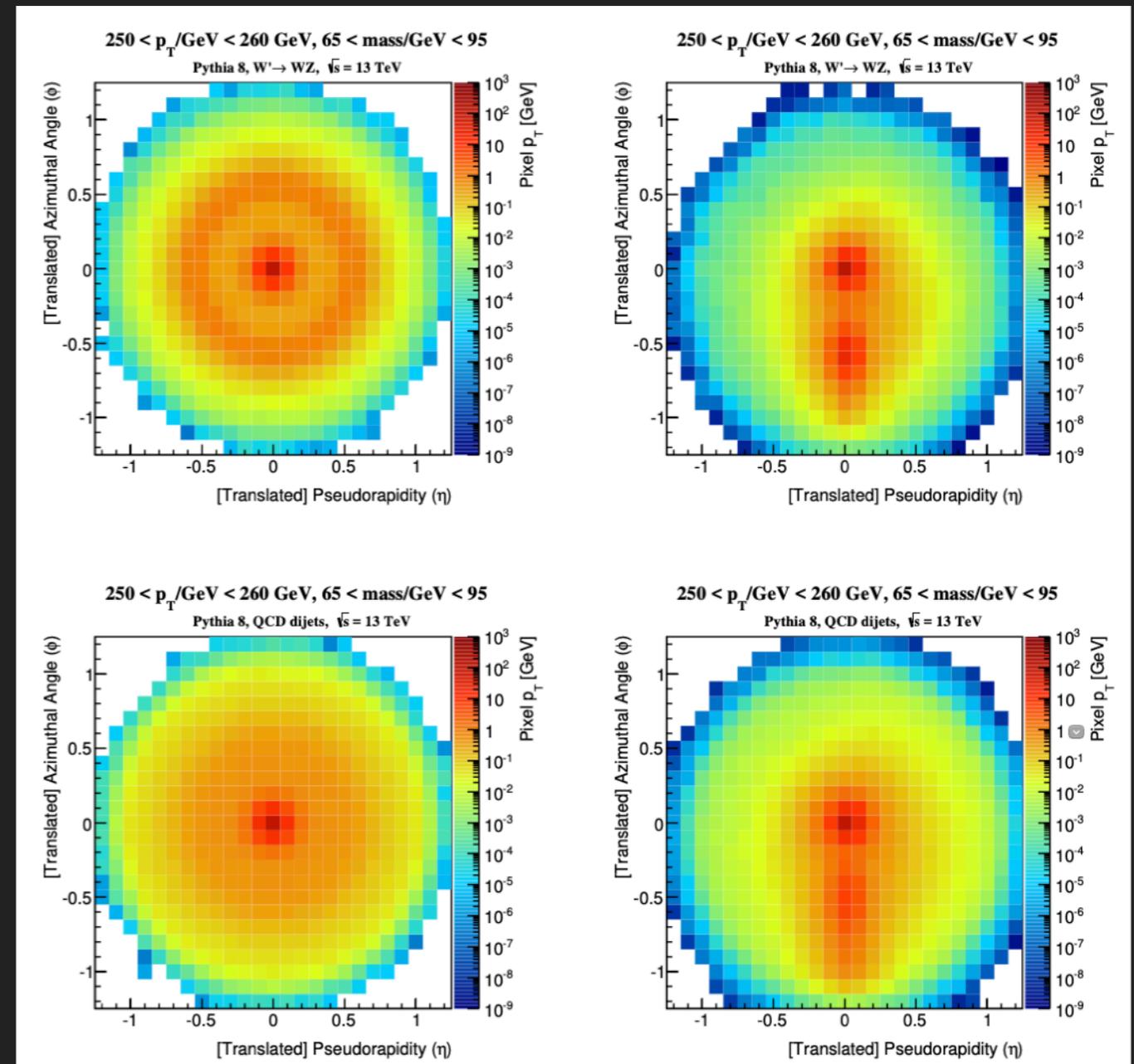
(de Oliveria et. al. 2017)



CNN EN EL LHC

(de Oliveria et. al. 2017)

- ▶ Imágenes de jets a partir de simulaciones en Pythia de W jets (arriba) y QCD jets (abajo) antes (izquierda) y después de pre procesamiento (derecha)
- ▶ Supera discriminadores, de W jets por encima de QCD jets, basados en variables físicas
 - ▶ e.g. masa del jet



CNN EN EL LHC

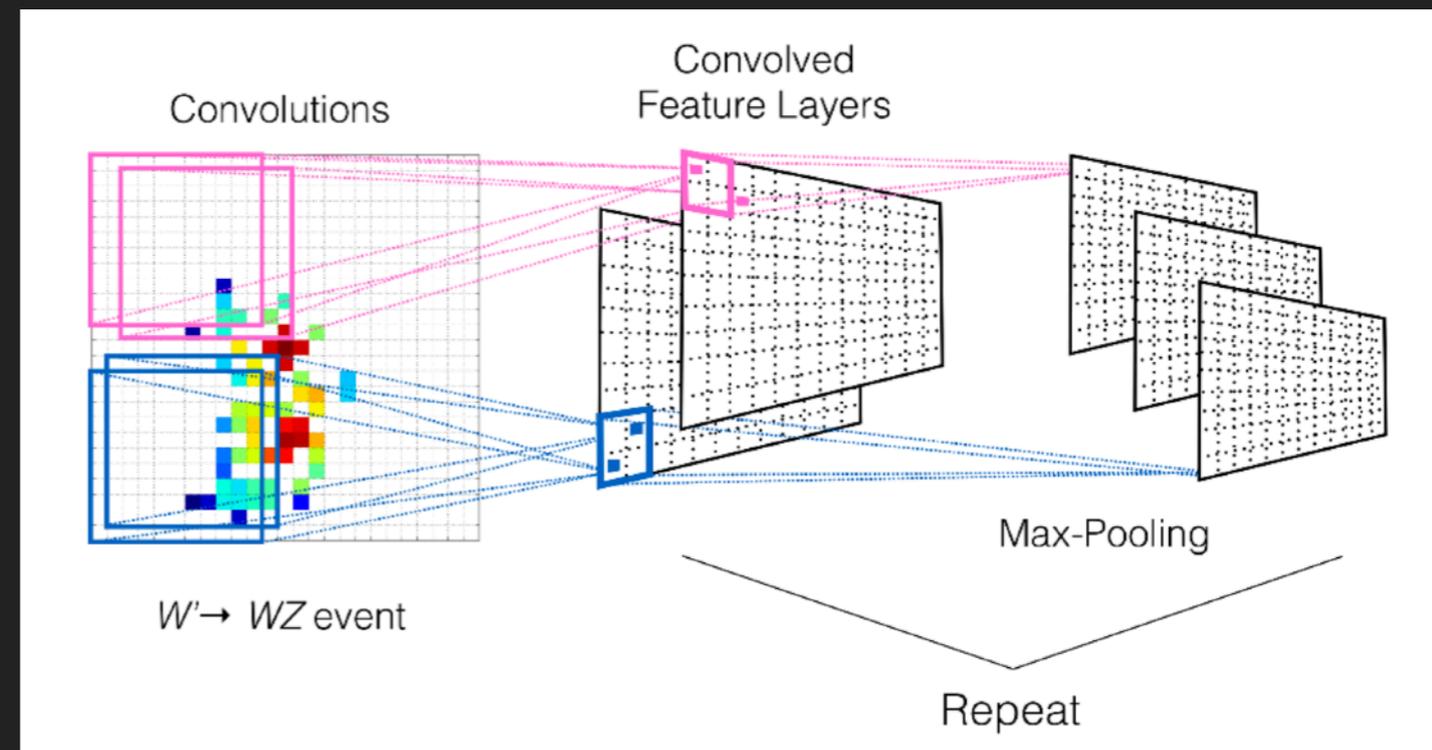
(de Oliveria et. al. 2017)

- ▶ Imágenes de jets a partir de simulaciones en Pythia de W jets (arriba) y QCD jets (abajo) antes (izquierda) y después de pre procesamiento (derecha)
- ▶ Supera discriminadores, de W jets por encima de QCD jets, basados en variables físicas
 - ▶ e.g. masa del jet

CNN EN EL LHC

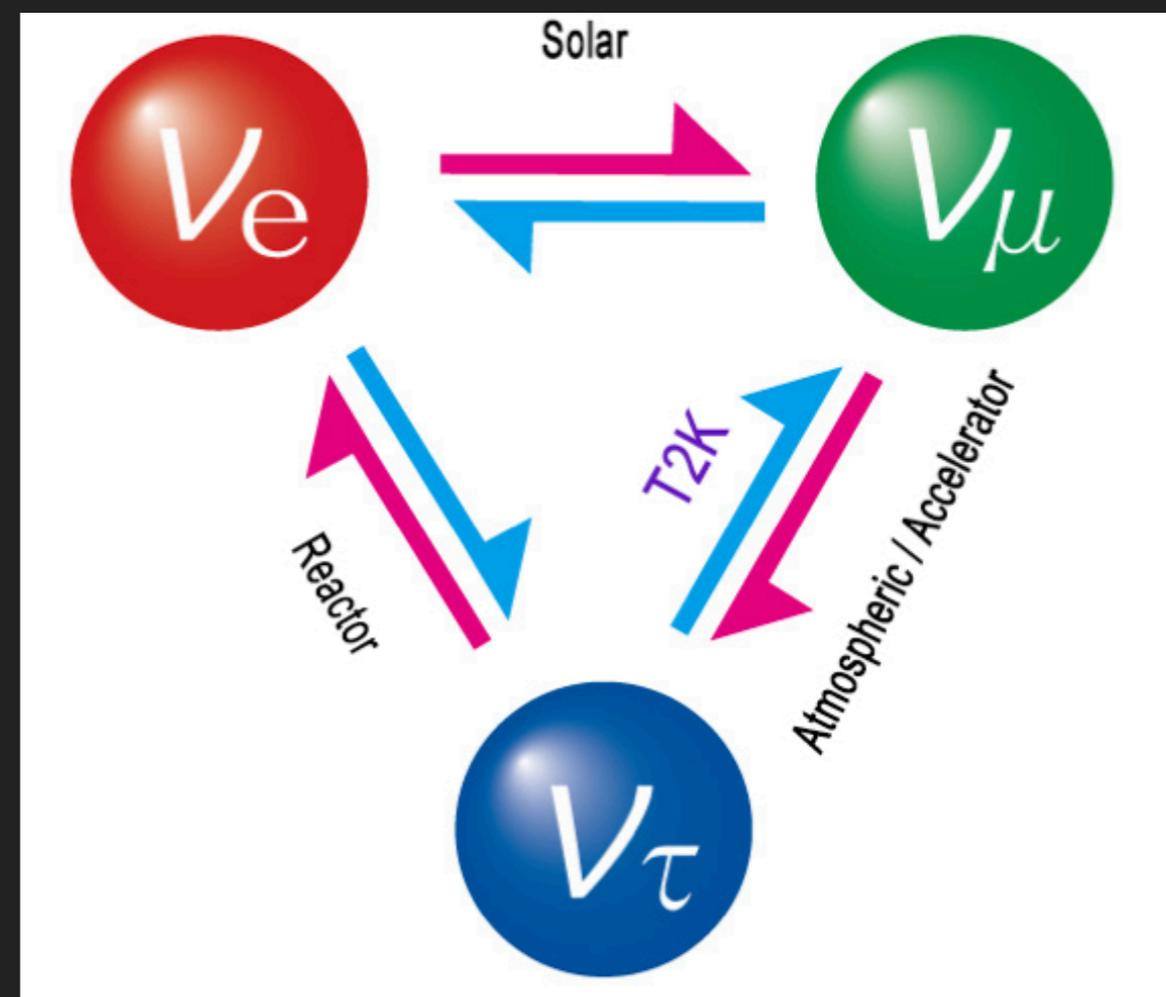
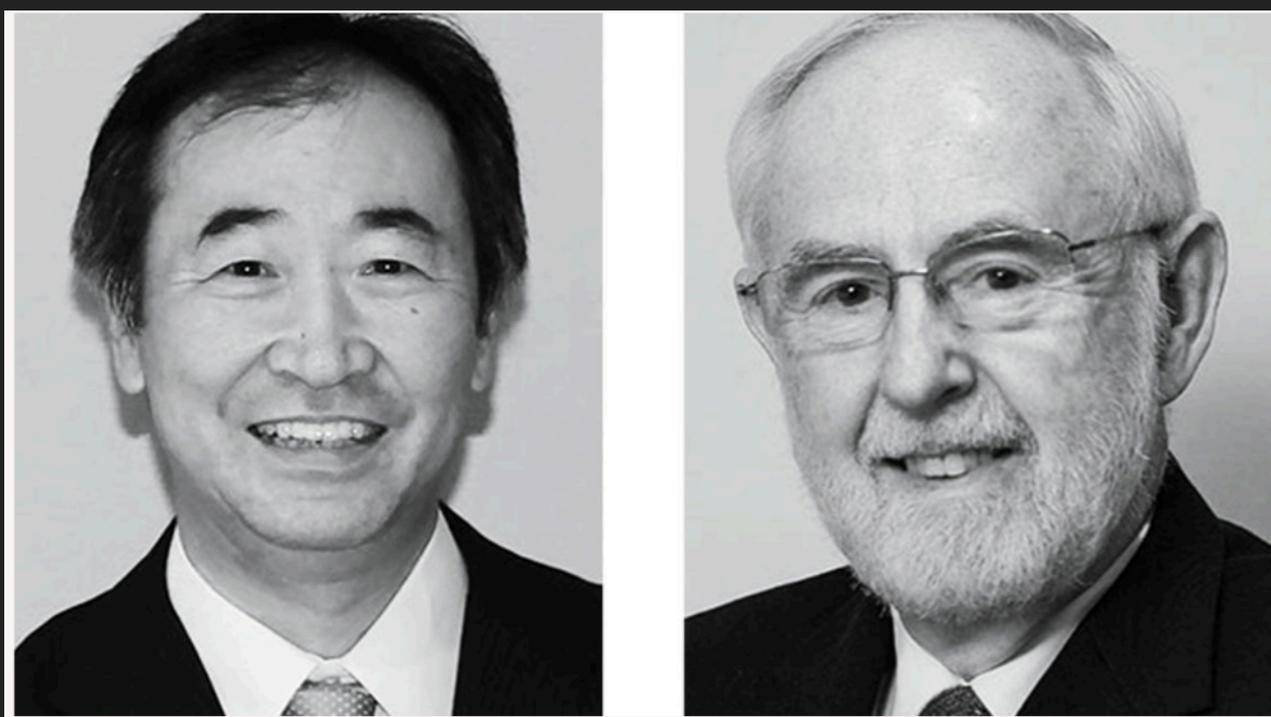
- ▶ Imágenes de jets a partir de simulaciones en Pythia de W jets (arriba) y QCD jets (abajo) antes (izquierda) y después de pre procesamiento (derecha)
- ▶ Supera discriminadores, de W jets por encima de QCD jets, basados en variables físicas
 - ▶ e.g. masa del jet

(de Oliveria et. al. 2017)



OSCILACIÓN DE NEUTRINOS

► Premio Nobel de Física 2015



Takaaki Kajita in Japan & Arthur B. McDonald in Canada

OTROS EXPERIMENTOS

MICROBOONE

MICROBOONE

- ▶ Fermilab

MICROBOONE

▶ Fermilab

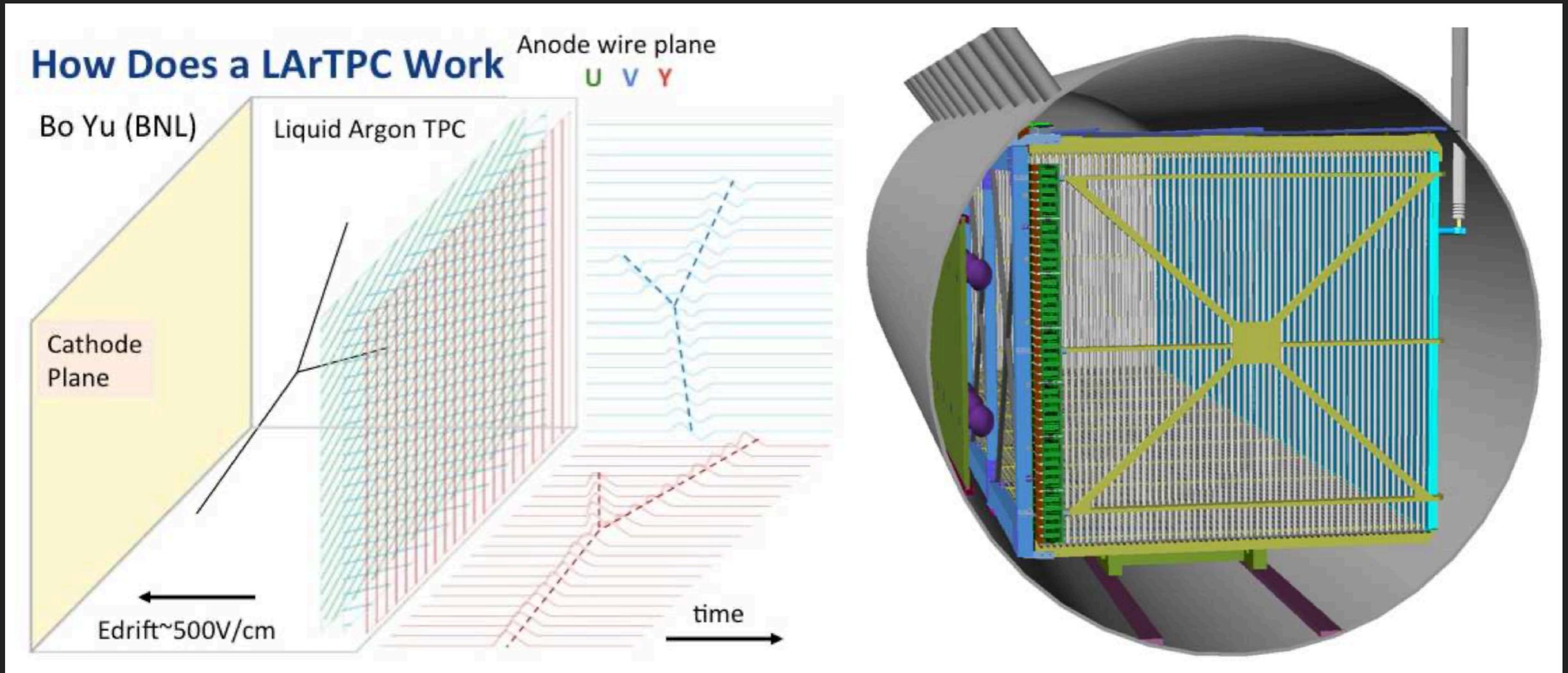


MICROBOONE

- ▶ Fermilab

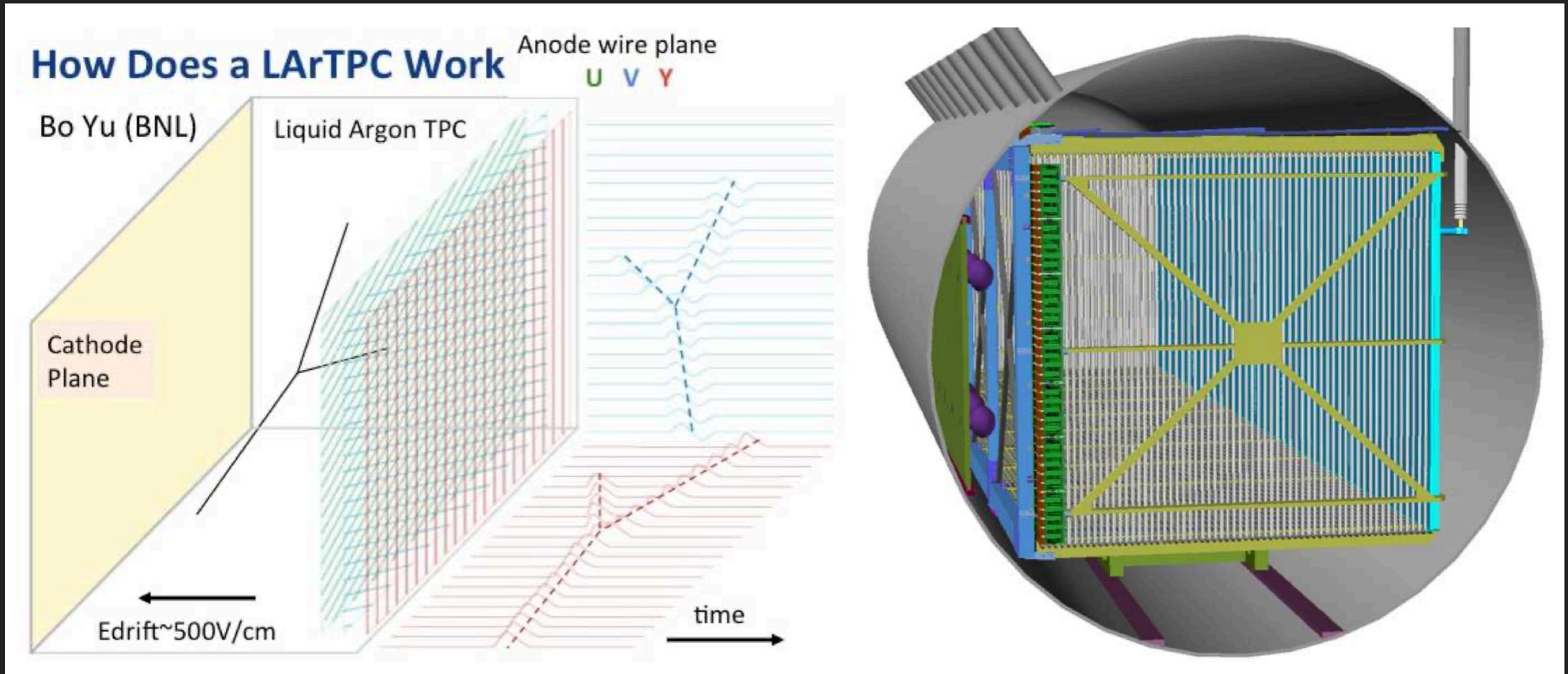
MICROBOONE

► Fermilab



MICROBOONE

- ▶ Fermilab
- ▶ 170 toneladas de Ar liquido

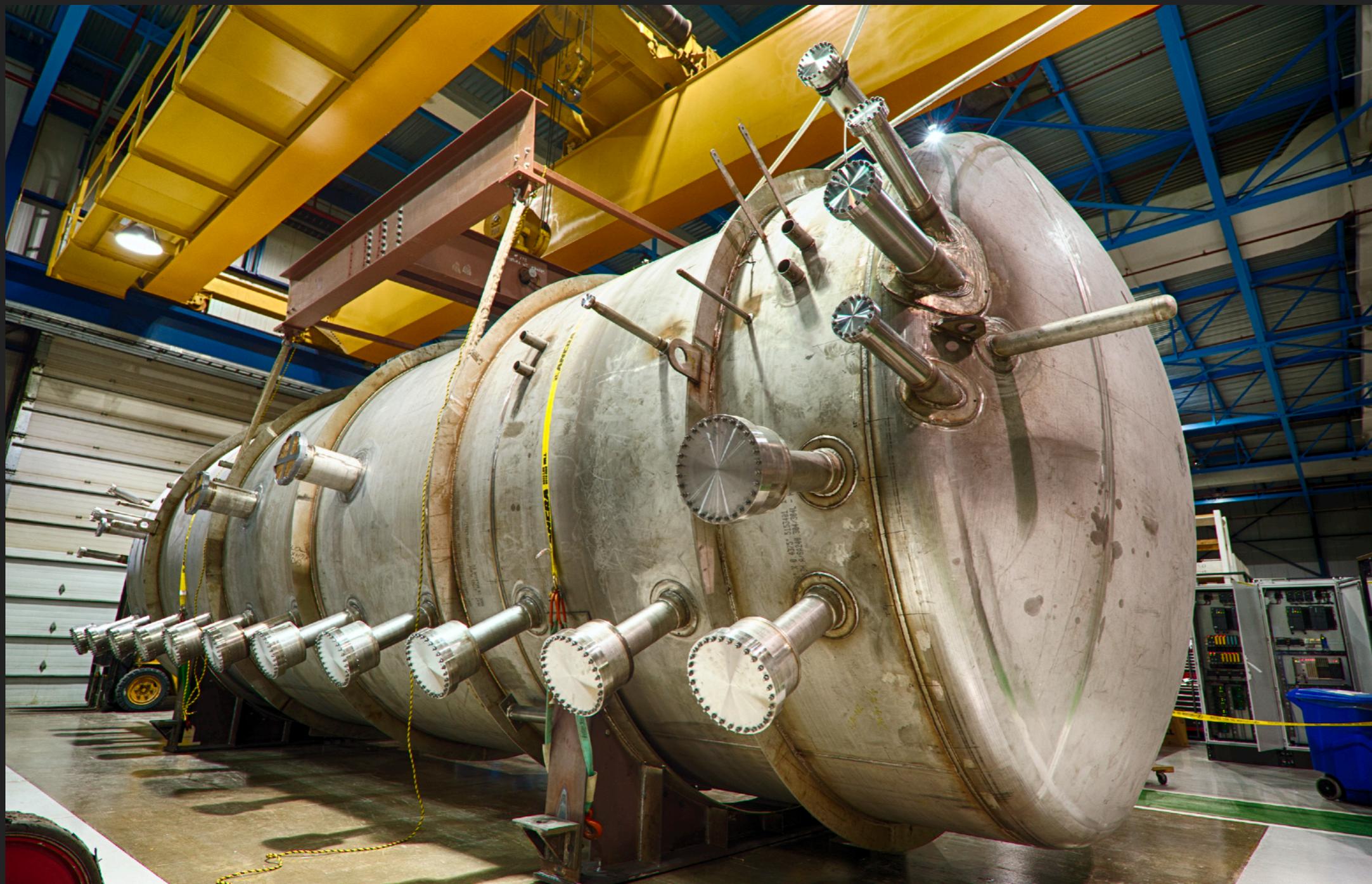


MICROBOONE

- ▶ Fermilab
- ▶ 170 toneladas de Ar líquido

MICROBOONE

► Fermilab



CNN EN MICROBOONE (FERMILAB)

(MicroBooNE Colaboration. 2017)

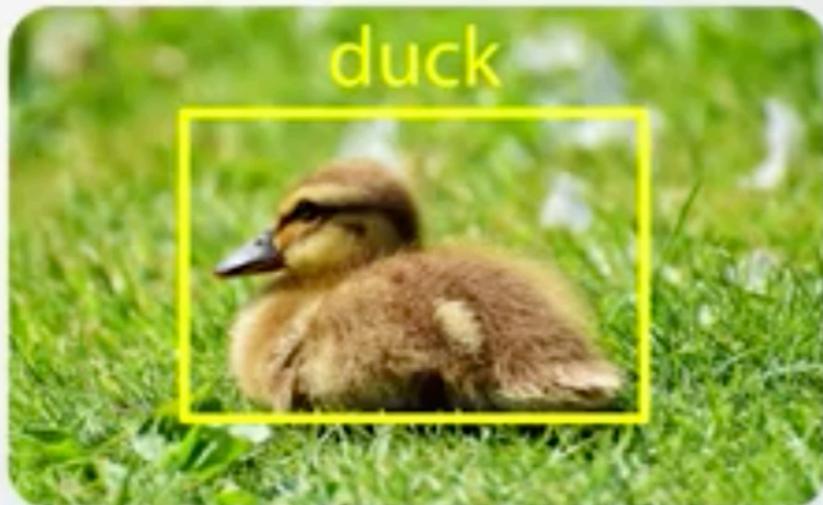
ν produce μ y π

CNN EN MICROBOONE (FERMILAB)

(MicroBooNE Colaboration. 2017)

ν produce μ y π

Object classification
+ localization:

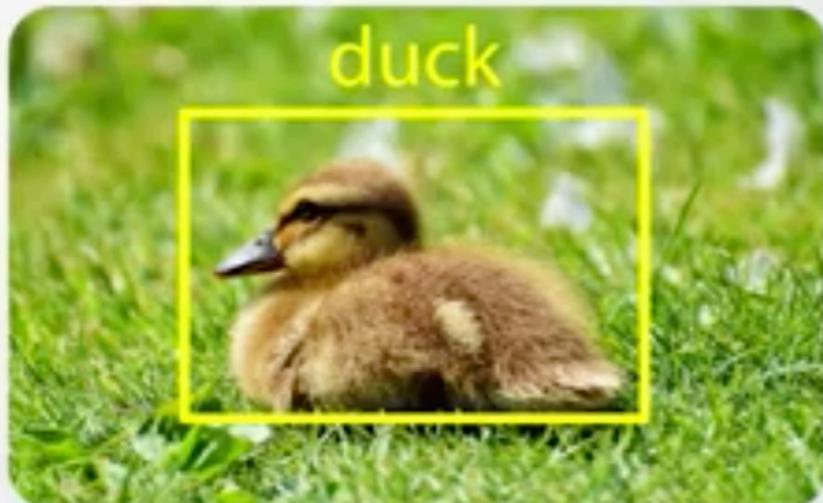


CNN EN MICROBOONE (FERMILAB)

(MicroBooNE Colaboration. 2017)

ν produce μ y π

Object classification
+ localization:



*J*inst

PUBLISHED BY IOP PUBLISHING FOR SISSA MEDIALAB

RECEIVED: *December 20, 2016*

REVISED: *February 11, 2017*

ACCEPTED: *February 28, 2017*

PUBLISHED: *March 14, 2017*

**Convolutional neural networks applied to neutrino events
in a liquid argon time projection chamber**

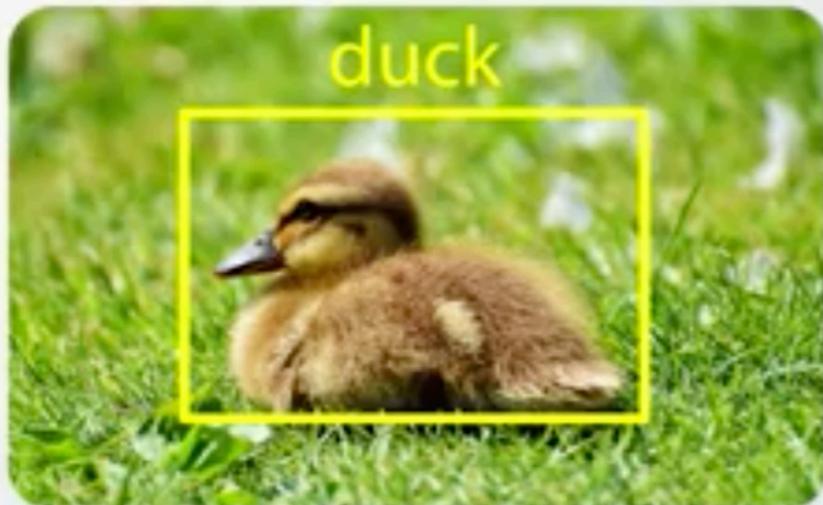
The MicroBooNE collaboration

CNN EN MICROBOONE (FERMILAB)

(MicroBooNE Colaboration. 2017)

ν produce μ y π

Object classification
+ localization:



Jinst

PUBLISHED BY IOP PUBLISHING FOR SISSA MEDIALAB

RECEIVED: December 20, 2016

REVISED: February 11, 2017

ACCEPTED: February 28, 2017

PUBLISHED: March 14, 2017

Convolutional neural networks applied to neutrino events in a liquid argon time projection chamber

The MicroBooNE collaboration

ABSTRACT: We present several studies of convolutional neural networks applied to data coming from the MicroBooNE detector, a liquid argon time projection chamber (LArTPC). The algorithms studied include the classification of single particle images, the localization of single particle and neutrino interactions in an image, and the detection of a simulated neutrino event overlaid with cosmic ray backgrounds taken from real detector data. These studies demonstrate the potential of convolutional neural networks for particle identification or event detection on simulated neutrino interactions. We also address technical issues that arise when applying this technique to data from a large LArTPC at or near ground level.

KEYWORDS: Analysis and statistical methods; Particle identification methods; Image filtering; Time projection chambers

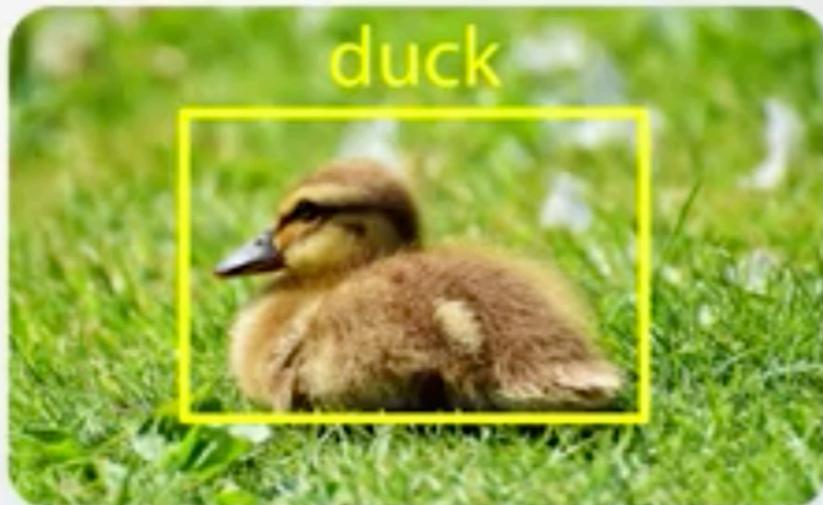
ARXIV ePRINT: [1611.05531](https://arxiv.org/abs/1611.05531)

CNN EN MICROBOONE (FERMILAB)

(MicroBooNE Colaboration. 2017)

ν produce μ y π

Object classification
+ localization:



Jinst

PUBLISHED BY IOP PUBLISHING FOR SISSA MEDIALAB

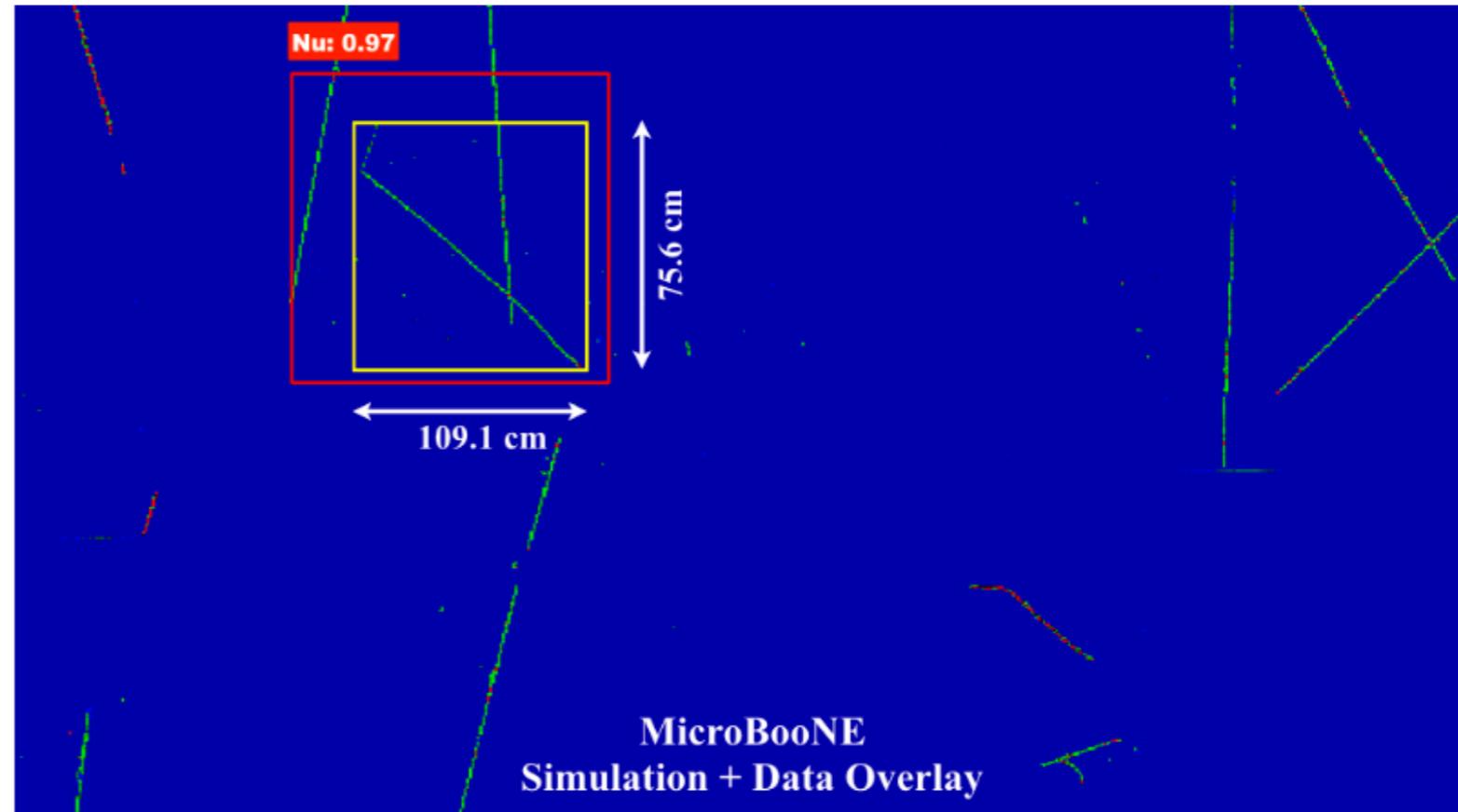
RECEIVED: December 20, 2016

REVISED: February 11, 2017

ACCEPTED: February 28, 2017

PUBLISHED: March 14, 2017

Convolutional neural networks applied to neutrino events



projection chambers

- ▶ “Aproximadamente 70% de los datos seleccionados por el trigger del Gran Colisionador de Hadrones son seleccionados por algoritmos de machine learning” - Kyle Cranmer (Carleo et. al. 2019)
- ▶ Ojo: cosas por mejorar
 - ▶ Explicabilidad de modelos de deep learning
 - ▶ Modelos robustos ante incertidumbres y sesgos (Simulaciones)
 - ▶ para subsiguiente implementación en experimentos
 - ▶ Integración de simetrías físicas a los modelos

APLICACIONES A CIENCIAS FÍSICAS

APLICACIONES A CIENCIAS FÍSICAS

- ▶ Química y materiales

APLICACIONES A CIENCIAS FÍSICAS

- ▶ Química y materiales
 - ▶ Exploración de superficies de energía libre (FES) de plegamiento de proteínas

APLICACIONES A CIENCIAS FÍSICAS

- ▶ Química y materiales
 - ▶ Exploración de superficies de energía libre (FES) de plegamiento de proteínas
- ▶ Física cuántica de muchos cuerpos

APLICACIONES A CIENCIAS FÍSICAS

- ▶ Química y materiales
 - ▶ Exploración de superficies de energía libre (FES) de plegamiento de proteínas
- ▶ Física cuántica de muchos cuerpos
 - ▶ Neural-Network Quantum States (NQS) -> representar $\Psi(\mathbf{r})$

APLICACIONES A CIENCIAS FÍSICAS

- ▶ Química y materiales
 - ▶ Exploración de superficies de energía libre (FES) de plegamiento de proteínas
- ▶ Física cuántica de muchos cuerpos
 - ▶ Neural-Network Quantum States (NQS) -> representar $\Psi(\mathbf{r})$
- ▶ Física Estadística aplicada a ML y vice versa

APLICACIONES A CIENCIAS FÍSICAS

- ▶ Química y materiales
 - ▶ Exploración de superficies de energía libre (FES) de plegamiento de proteínas
- ▶ Física cuántica de muchos cuerpos
 - ▶ Neural-Network Quantum States (NQS) -> representar $\Psi(\mathbf{r})$
- ▶ Física Estadística aplicada a ML y vice versa
 - ▶ Maquinas restringidas de Boltzmann/detección de transiciones de fase

APLICACIONES A CIENCIAS FÍSICAS

- ▶ Química y materiales
 - ▶ Exploración de superficies de energía libre (FES) de plegamiento de proteínas
- ▶ Física cuántica de muchos cuerpos
 - ▶ Neural-Network Quantum States (NQS) -> representar $\Psi(\mathbf{r})$
- ▶ Física Estadística aplicada a ML y vice versa
 - ▶ Maquinas restringidas de Boltzmann/detección de transiciones de fase
- ▶ Computación cuántica

APLICACIONES A CIENCIAS FÍSICAS

- ▶ Química y materiales
 - ▶ Exploración de superficies de energía libre (FES) de plegamiento de proteínas
- ▶ Física cuántica de muchos cuerpos
 - ▶ Neural-Network Quantum States (NQS) -> representar $\Psi(\mathbf{r})$
- ▶ Física Estadística aplicada a ML y vice versa
 - ▶ Maquinas restringidas de Boltzmann/detección de transiciones de fase
- ▶ Computación cuántica
 - ▶ Aprendizaje de distribuciones sobre medidas cuánticas (Quantum State Tomography)

BIBLIOGRAFÍA

- ▶ Carleo, Giuseppe, et al. "Machine learning and the physical sciences." *Reviews of Modern Physics* 91.4 (2019): 045002.
- ▶ Guest, Dan, Kyle Cranmer, and Daniel Whiteson. "Deep learning and its application to LHC physics." *Annual Review of Nuclear and Particle Science* 68 (2018): 161-181.
- ▶ Elwood, Adam, and Dirk Krücker. "Direct optimisation of the discovery significance when training neural networks to search for new physics in particle colliders." *arXiv preprint arXiv:1806.00322* (2018).
- ▶ Sahin, Mehmet Özgür, Dirk Krücker, and I-A. Melzer-Pellmann. "Performance and optimization of support vector machines in high-energy physics classification problems." *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 838 (2016): 137-146.
- ▶ <https://www.coursera.org/learn/hadron-collider-machine-learning/>

BIBLIOGRAFÍA CONTINUADA

- ▶ de Oliveira, Luke, et al. "Jet-images–deep learning edition." *Journal of High Energy Physics* 2016.7 (2016): 69.
- ▶ Farina, Marco, Yuichiro Nakai, and David Shih. "Searching for new physics with deep autoencoders." *arXiv preprint arXiv:1808.08992* (2018).
- ▶ Shimmin, Chase, et al. "Decorrelated jet substructure tagging using adversarial neural networks." *Physical Review D* 96.7 (2017): 074034.
- ▶ Radovic, Alexander, et al. "Machine learning at the energy and intensity frontiers of particle physics." *Nature* 560.7716 (2018): 41-48.
- ▶ Acciarri, R., et al. "Convolutional neural networks applied to neutrino events in a liquid argon time projection chamber." *Journal of instrumentation* 12.03 (2017): P03011.
- ▶ Soderberg, Mitchell, and MicroBooNE Collaboration. "MicroBooNE: A New Liquid Argon Time Projection Chamber Experiment." *AIP Conference Proceedings*. Vol. 1189. No. 1. American Institute of Physics, 2009.

AUTOENCODERS

- ▶ Red Neuronal Pre-alimentada (Feed-forward)
- ▶ Cuello de Botella Informatico

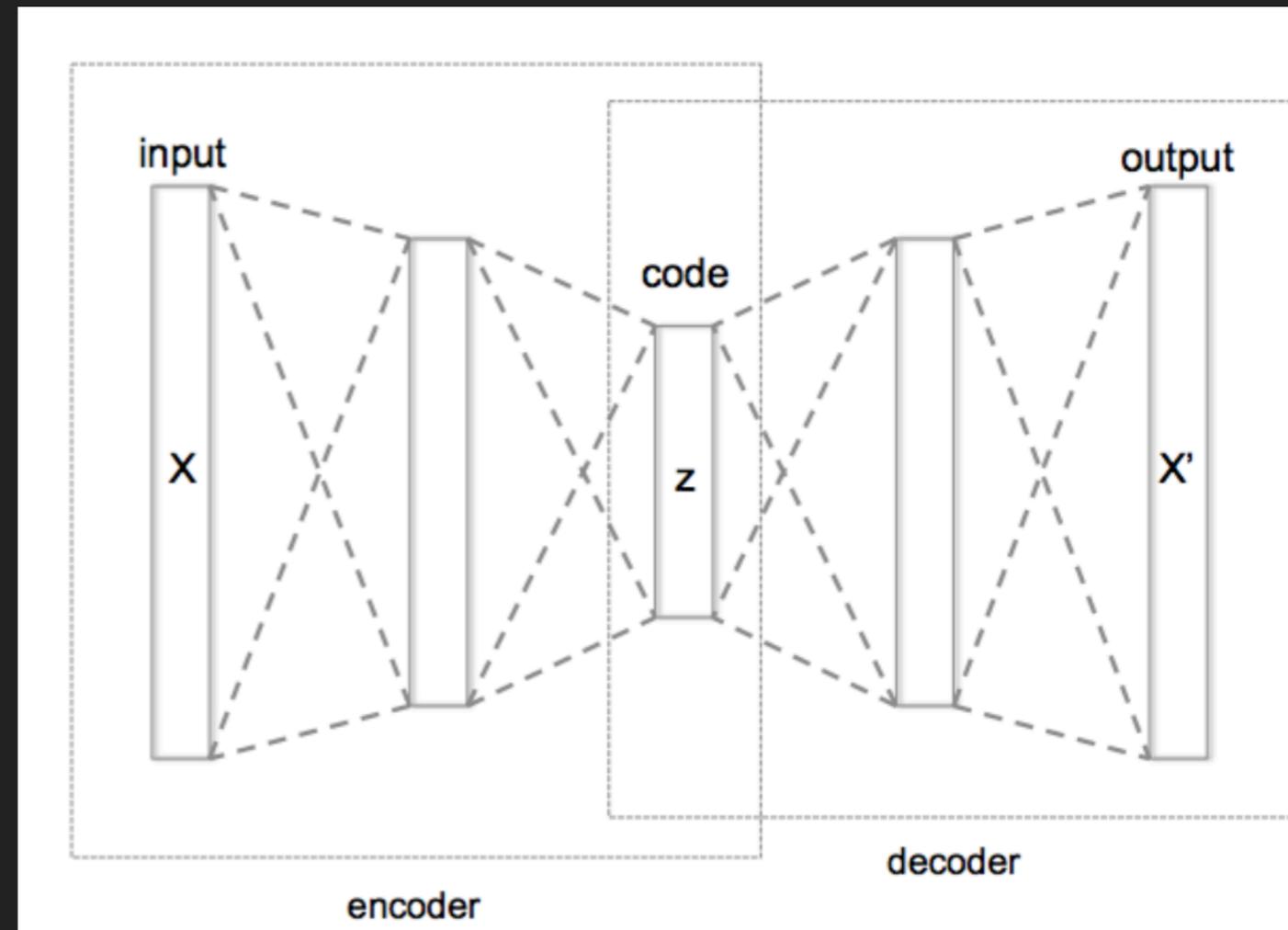
- ▶ **Encoder:**

$X \rightarrow \text{Dense}(d \text{ unidades}) \rightarrow \text{code}$

- ▶ **Decoder:**

$\text{code} \rightarrow \text{Dense}(m \text{ unidades}) \rightarrow X$

- ▶ Minimizar MSE entre input y output



AUTOENCODERS

- ▶ Red Neuronal Pre-alimentada (Feed-forward)
- ▶ Cuello de Botella Informatico

- ▶ **Encoder:**

$X \rightarrow \text{Dense}(d \text{ unidades}) \rightarrow \text{code}$

- ▶ **Decoder:**

$\text{code} \rightarrow \text{Dense}(m \text{ unidades}) \rightarrow X$

- ▶ Minimizar MSE entre input y output

AUTOENCODERS

- ▶ Red Neuronal Pre-alimentada (Feed-forward)

- ▶ Cuello de Botella Informatico

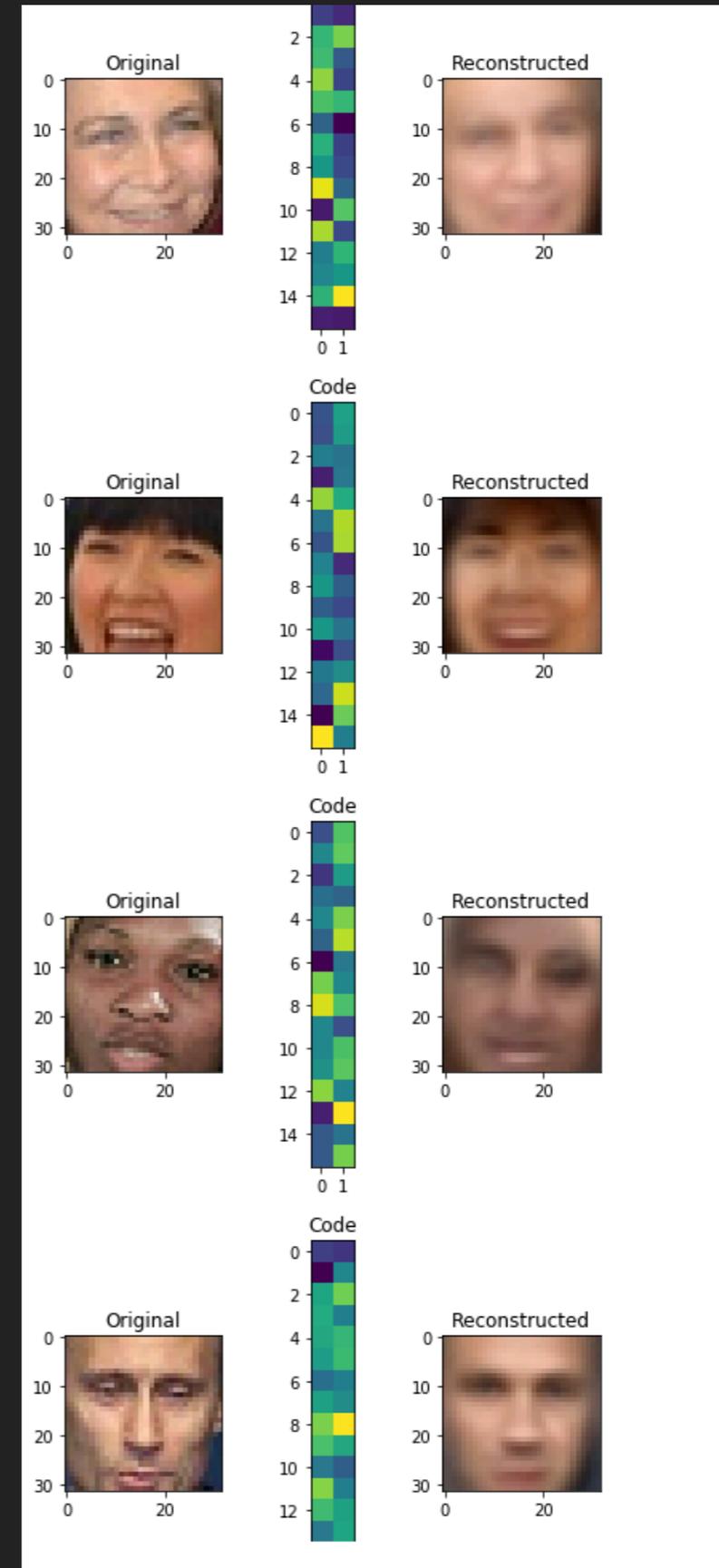
- ▶ **Encoder:**

$X \rightarrow \text{Dense}(d \text{ unidades}) \rightarrow \text{code}$

- ▶ **Decoder:**

$\text{code} \rightarrow \text{Dense}(m \text{ unidades}) \rightarrow X$

- ▶ Minimizar MSE entre input y output



AUTOENCODERS

- ▶ Red Neuronal Pre-alimentada (Feed-forward)

- ▶ Cuello de Botella Informatico

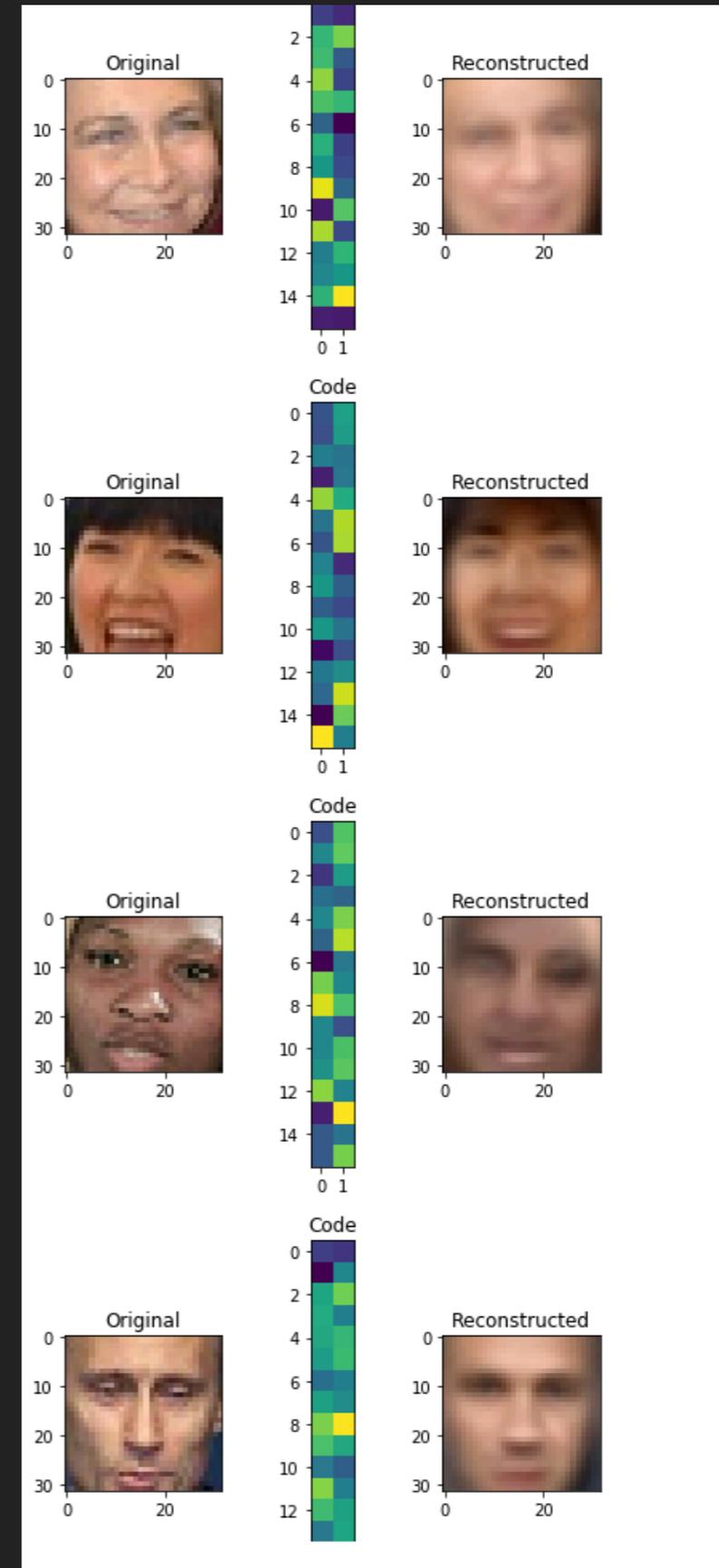
- ▶ **Encoder:**

$X \rightarrow \text{Dense}(d \text{ unidades}) \rightarrow \text{code}$

- ▶ **Decoder:**

$\text{code} \rightarrow \text{Dense}(m \text{ unidades}) \rightarrow X$

- ▶ Minimizar MSE entre input y output



DENOISING AUTOENCODER

```
s = reset_tf_session()

# we use bigger code size here for better quality
encoder, decoder = build_deep_autoencoder(IMG_SHAPE, code_size=512)
assert encoder.output_shape[1:]==(512,), "encoder must output a code of required size"

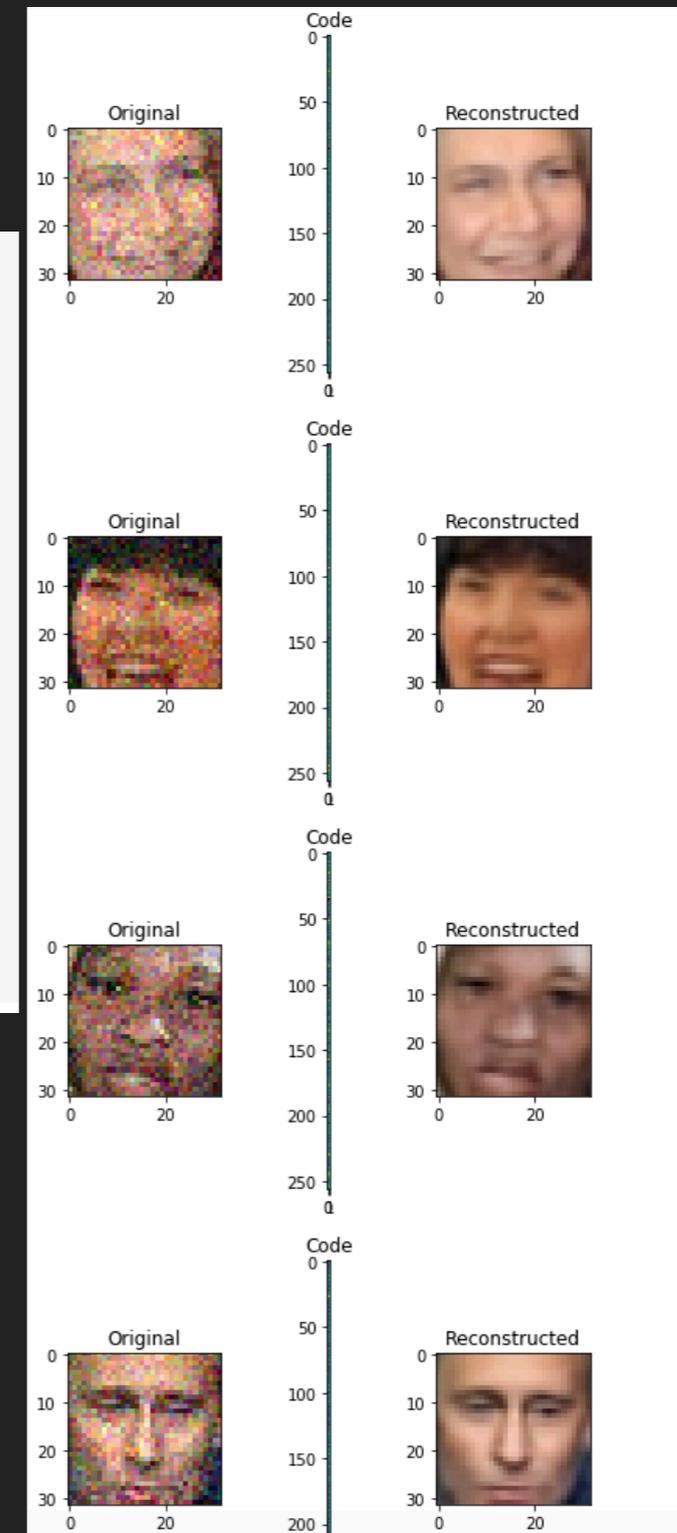
inp = L.Input(IMG_SHAPE)
code = encoder(inp)
reconstruction = decoder(code)

autoencoder = keras.models.Model(inp, reconstruction)
autoencoder.compile('adamax', 'mse')

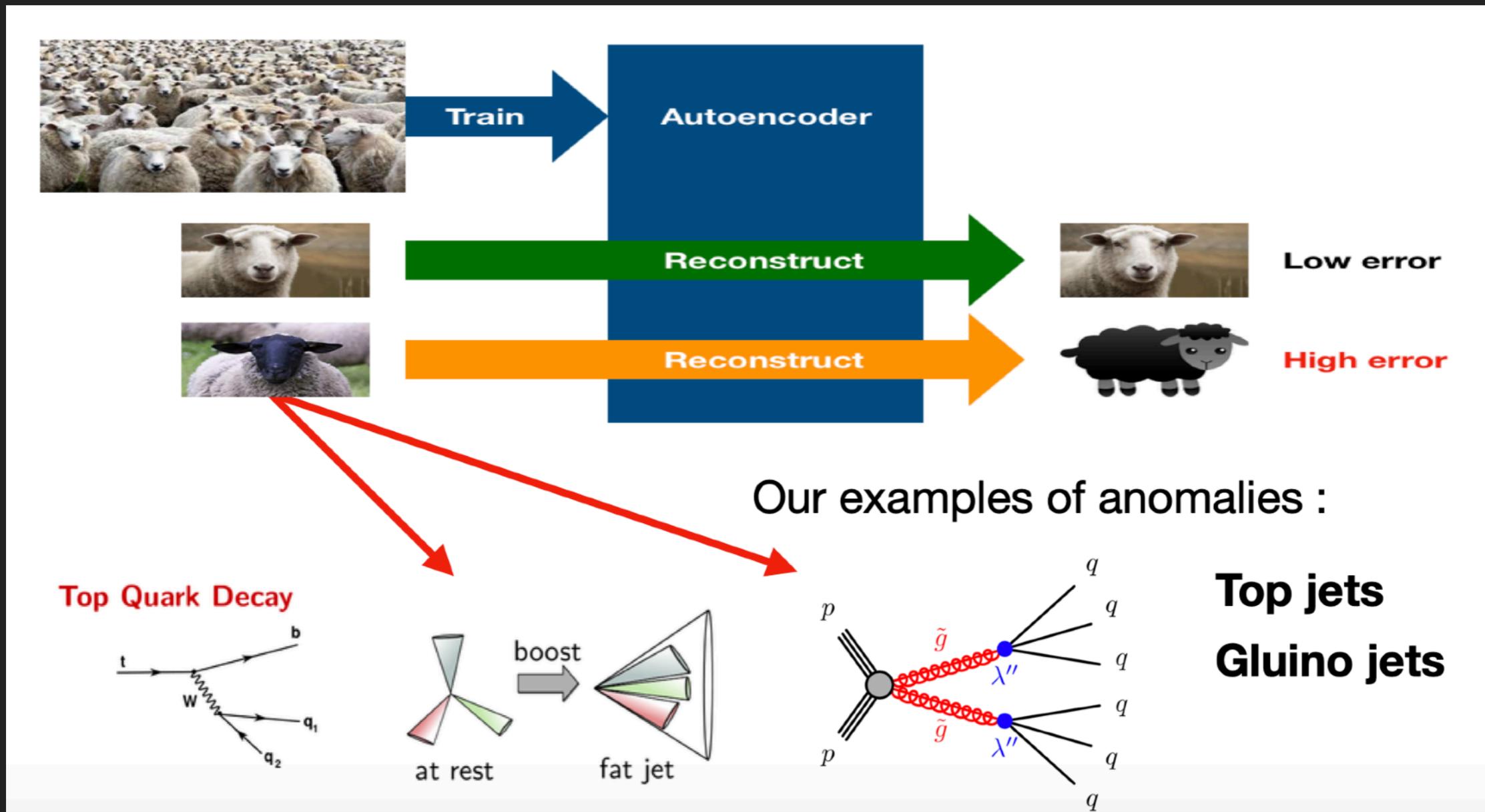
for i in range(25):
    print("Epoch %i/25, Generating corrupted samples.."%(i+1))
    X_train_noise = apply_gaussian_noise(X_train)
    X_test_noise = apply_gaussian_noise(X_test)

    # we continue to train our model with new noise-augmented data
    autoencoder.fit(x=X_train_noise, y=X_train, epochs=1,
                  validation_data=[X_test_noise, X_test],
                  callbacks=[keras_utils.TqdmProgressCallback()],
                  verbose=0)
```

```
1 X_test_noise = apply_gaussian_noise(X_test)
2 denoising_mse = autoencoder.evaluate(X_test_noise, X_test, verbose=0)
3 print("Denoising MSE:", denoising_mse)
4 for i in range(3):
5     img = X_test_noise[i]
6     visualize(img, encoder, decoder)
```



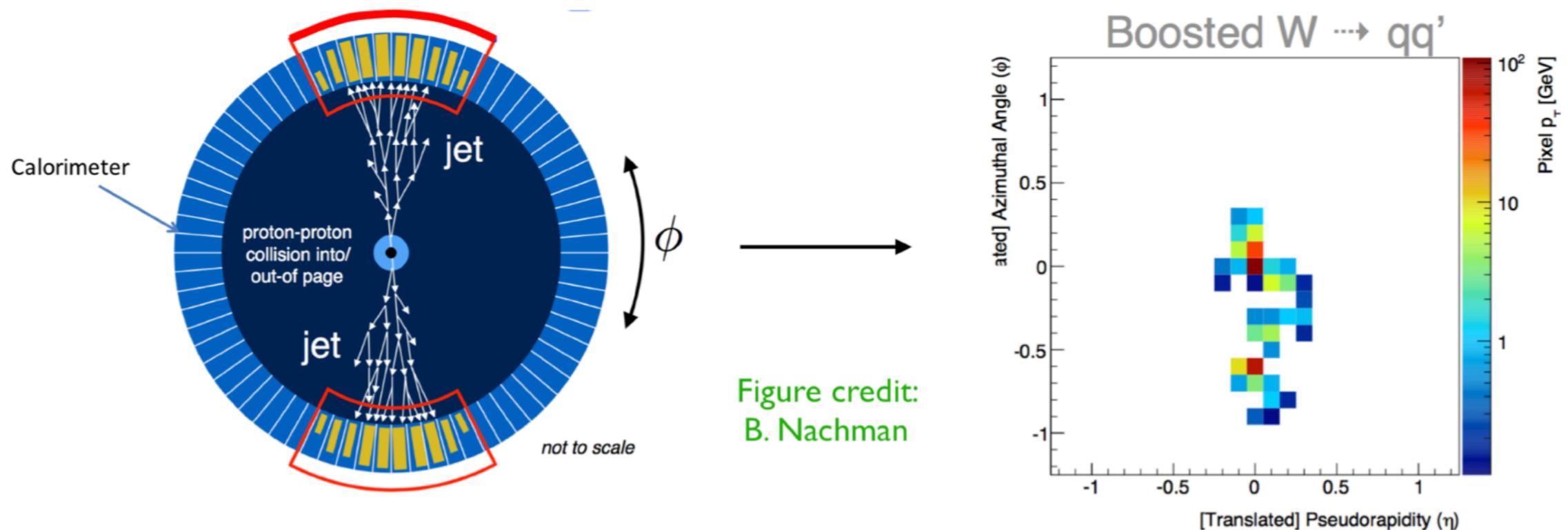
AUTOENCODERS EN DETECCIÓN DE ANOMALIAS



AUTOENCODERS EN DETECCIÓN DE ANOMALIAS

AUTOENCODERS EN DETECCIÓN DE ANOMALIAS

Focus on jet images (2D of eta and phi) as inputs to autoencoder.



ENTRENAR REDES NEURONALES CON SIGNIFICACIÓN DE DESCUBRIMIENTO

ENTRENAR REDES NEURONALES CON SIGNIFICACIÓN DE DESCUBRIMIENTO

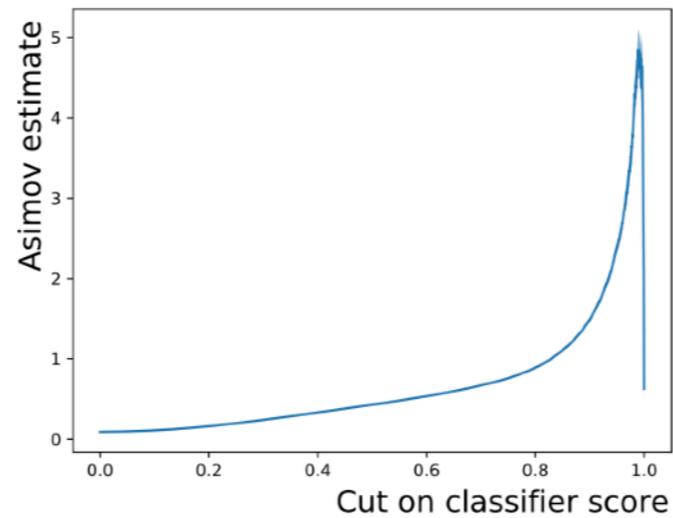
- ▶ “Maximización de la correcta clasificación de señales, a la vez minimizando la incorrecta clasificación de eventos de background” - (Elwood & Krücker 2019)

ENTRENAR REDES NEURONALES CON SIGNIFICACIÓN DE DESCUBRIMIENTO

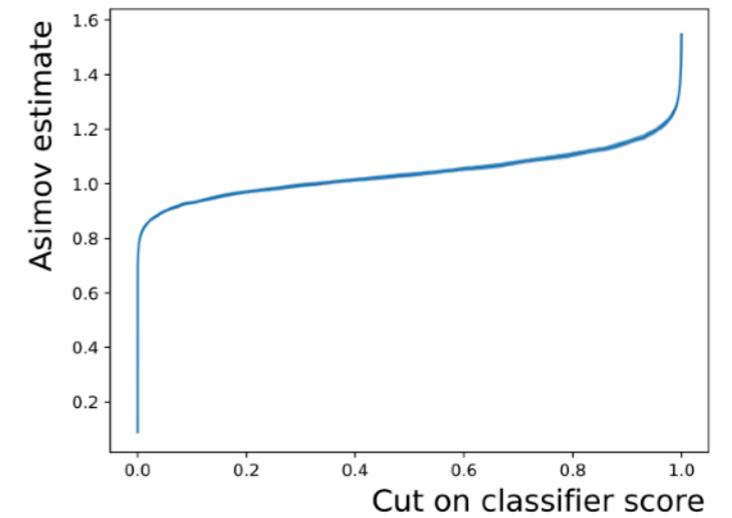
- ▶ “Maximización de la correcta clasificación de señales, a la vez minimizando la incorrecta clasificación de eventos de background” - (Elwood & Krücker 2019)
- ▶ búsqueda de SUSY, datos simulados de producción de stops

ENTRENAR REDES NEURONALES CON SIGNIFICACIÓN DE DESCUBRIMIENTO

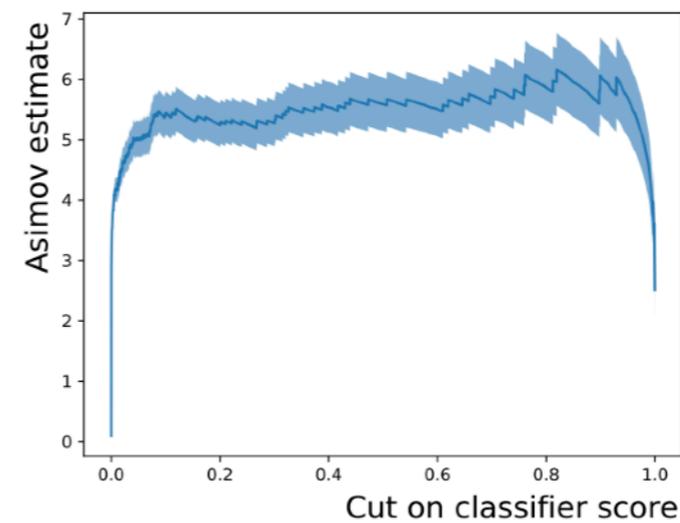
- ▶ “Maximización de la correcta clasificación de señales, a la vez minimizando la incorrecta clasificación de eventos de background” - (Elwood & Krücker 2019)
- ▶ búsqueda de SUSY, datos simulados de producción de stops



(a) Binary cross entropy



(b) $\ell_{s/\sqrt{s+b}}$



(c) ℓ_{Asimov}

