

Universidad Autónoma de Ciudad Juárez

INSTITUTO DE INGENIERÍA Y TECNOLOGÍA
Departamento de Física y Matemáticas



Entrenamiento de un modelo AutoEncoder (AE) para detección de eventos anómalos usando datos del experimento CMS del CERN.

Proyecto de Titulación que presenta:
Miguel Ángel TERRAZAS MARTÍNEZ

Como requisito parcial para obtener el título de:
INGENIERO FÍSICO

Asesor:
Dr. Alfredo Martín CASTAÑEDA HERNÁNDEZ

CD. JUÁREZ, CHIHUAHUA
MAYO DE 2025

Universidad Autónoma de Ciudad Juárez
Departamento de Física y Matemáticas

Proyecto de Titulación. Memoria

**Entrenamiento de un modelo AutoEncoder (AE)
para detección de eventos anómalos usando datos
del experimento CMS del CERN.**

Autor: Miguel Ángel TERRAZAS MARTÍNEZ

Proyecto de titulación presentado a la Academia de Ingeniería Física como requisito
para obtener el título de Ingeniero Físico por la Universidad Autónoma de Ciudad
Juárez

Programa de licenciatura
Ingeniería Física

Asesor: Dr. Alfredo Martín CASTAÑEDA HERNANDEZ

CIUDAD JUÁREZ, CHIHUAHUA, MAYO DE 2025

Declaración de Originalidad

Yo, Miguel Ángel TERRAZAS MARTÍNEZ, declaro que el material contenido en este documento fue redactado por mí y no es un plagio total o parcial, ni ha sido usado para obtener otro título o reconocimiento en otra institución de educación superior.




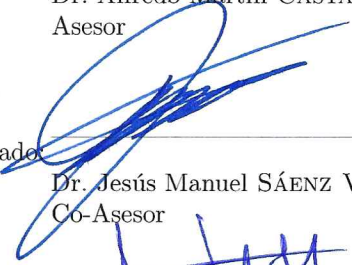
Miguel Ángel TERRAZAS MARTÍNEZ

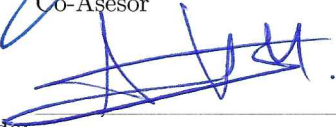
Entrenamiento de un modelo AutoEncoder (AE) para detección de eventos anómalos usando datos del experimento CMS del CERN.


Los miembros del comité evaluador que aprobaron el manuscrito del proyecto de titulación de:

Miguel Ángel TERRAZAS MARTÍNEZ

Aprobado: 
Dr. Alfredo Martín CASTAÑEDA HERNANDEZ
Asesor

Aprobado: 
Dr. Jesús Manuel SÁENZ VILLELA
Co-Asesor

Aprobado: 
Dr. Héctor Alejandro TREJO MANDUJANO
Sinodal

Aprobado: 
Dr. Gildardo RIVAS VALLES
Sinodal

Aprobado: _____
Dra. Imelda OLIVAS ARMENDÁRIZ
Profesor de la materia

Resumen

Se desarrolló y ajustó un autoencoder denso para detectar eventos inusuales con tres o más leptones en las simulaciones de colisiones protón-protón del experimento CMS, persiguiendo la identificación de desviaciones respecto al Modelo Estándar en topologías trileptónicas, donde la elevada multiplicidad de observables y la gran tasa de producción dificultan el uso de métodos estadísticos convencionales. El flujo de trabajo comprendió la selección de muestras Monte Carlo de fondo (WZ , ZZ , Drell–Yan) y de una señal de referencia $W' \rightarrow WZ \rightarrow 3\ell$ de 2 TeV; el filtrado y la reducción de los ficheros NanoAOD para conservar 5 variables cinemáticas esenciales; la conversión a formato HDF5 para acelerar la carga masiva; y el entrenamiento del modelo con una partición independiente para validación. La función de pérdida mostró convergencia estable en menos de veinte épocas y la curva ROC arrojó un área bajo la curva de 0.99. Al definir un umbral de reconstrucción $\tau = 1300$ (percentil 95 del fondo) se retuvo el 75 % de la señal con una tasa de falsos positivos inferior al 0.2 %. La distribución de los errores reveló colas bien separadas, mientras que proyecciones tanto lineales (PCA) como no lineales (t -SNE) del espacio latente situaron los eventos candidatos en regiones periféricas distantes de la nube dominada por el fondo. Estos resultados acreditan la eficacia parcial del autoencoder como herramienta en la búsqueda de nueva física y sientan las bases para su aplicación directa a datos reales del CMS.

Agradecimientos

Quiero comenzar expresando mi agradecimiento a mi familia; a mis padres, Lizette y Jesús, por su paciencia y el cariño con el que me han acompañado en esta etapa; a mis hermanos, Sofía y Leonardo, por sus consejos y su constante apoyo; y a mi abuela Alicia, por todo el cariño que siempre me ha brindado. Gracias por estar a mi lado en cada momento y por motivarme a seguir adelante.

A mis mejores amigos Humberto, Iñaki, Marcelo, Javier, Rafael, Daniel, Horacio, Abdiel y Delgadillo, gracias por ofrecerme su apoyo emocional, por estar a mi lado en los momentos complicados y por recordarme la importancia de tomarme un respiro cuando más lo necesitaba.

A mis compañeros y amigos de la carrera, gracias por su disposición y compañía en el transcurso de los últimos semestres.

Al Dr. Castañeda, por darme su atención, por dedicarme su tiempo y por el apoyo para la realización del proyecto.

Finalmente, agradezco al servidor ACARUS de la Universidad de Sonora (UNISON), cuya disponibilidad fue de alta importancia para el procesamiento de datos y la ejecución de los programas empleados en el estudio.

Índice general

Índice de figuras	XIV
Índice de cuadros	XV
1. Introducción	1
1.1. Contexto	1
1.2. Antecedentes	1
1.3. Planteamiento del problema	2
1.4. Solución propuesta	2
1.5. Objetivo general	3
1.6. Objetivos específicos	3
1.7. Justificación	3
1.8. Impactos	3
1.9. Alcances y delimitaciones	3
1.9.1. Alcances	3
1.9.2. Delimitaciones	4
2. Marco Teórico	5
2.1. Introducción	5
2.2. Física de altas energías	5
2.2.1. El Modelo Estándar	5
2.2.2. Búsqueda física mas allá del Modelo Estándar	7
2.3. Machine learning	8
2.3.1. Redes neuronales	10
2.3.2. Autoencoders	11
2.3.3. Métricas de validación de anomalías	13
2.4. Deep learning aplicado a física de altas energías	13
2.4.1. Preprocesamiento de datos en HEP	14
2.4.2. Comparación con métodos clásicos	14
3. Metodología	15
3.1. Introducción	15
3.2. Selección de muestras	15
3.3. Procesamiento de Muestras	16
3.3.1. Reducción de muestras (skimming) y selección de variables	16
3.3.2. Mezclado de eventos aleatorio	18
3.3.3. Conversión a formato hdf5	18
3.4. Entrenamiento de modelo autoencoder	18

4. Resultados y discusión	21
4.1. Introducción	21
4.2. Gráficas de entrenamiento	21
5. Conclusiones	28
A. Comparación ruido y señal	31
B. Secciones de código	37
B.1. Bloque central del AutoEncoder	37
B.2. Configuración del Trainer y callbacks	38

Índice de figuras

2.1. Diagrama simplificado del Modelo Estándar de física de partículas. Se muestran los seis quarks (<i>up</i> , <i>down</i> , <i>charm</i> , <i>strange</i> , <i>top</i> , <i>bottom</i>), los seis leptones (electrón, muón, tau y sus neutrinos), los bosones gauge (γ , W^\pm , Z^0 , gluones) y el bosón de Higgs (H^0) [13].	6
2.2. Diagrama de Feynman de la topología de estudio, en este caso representando la producción de una partícula W' , su decaimiento a WZ y finalmente 3 leptones.	9
3.1. Distribución de <code>MET_pt</code> para eventos de ruido (SM) y señal.	17
3.2. Matriz de correlación entre variables en la muestra de fondo.	17
3.3. Diagrama del AutoEncoder empleado. El “cuello de botella” de 16 neuronas obliga a la red a retener en z sólo la información crucial de cada evento.	19
4.1. Curva de pérdida <i>train / validation</i> . La convergencia y la pequeña separación entre ambas curvas evidencian la ausencia de sobreajuste significativo durante las últimas épocas.	22
4.2. Curva ROC para la detección de anomalías mediante el error de reconstrucción. El área bajo la curva $AUC = 0,99$ demuestra alta capacidad de separación entre señal y fondo.	23
4.3. PCA del espacio latente. El fondo (azul) se agrupa en torno al origen, mientras que la señal (rojo) se dispersa hacia valores positivos de PC1 y PC2, lo que revela la presencia de componentes lineales que distinguen a los eventos anómalos.	24
4.4. t-SNE del espacio latente. El fondo (morado) ocupa la mayor parte del espacio latente; los eventos de señal (amarillo) forman un racimo compacto en una región poco poblada, confirmando la capacidad del AE para aislar configuraciones atípicas.	25
4.5. Histograma del error de reconstrucción. La señal W' (naranja) se desplaza a valores de error mayores que el fondo (azul), indicando que el AE reconstruye peor los eventos anómalos.	26
4.6. Dispersión del error de reconstrucción por evento. Arriba: escala completa donde la señal (rojo) sobresale de acuerdo al umbral establecido	26
4.7. Ampliación de la banda donde se puede apreciar la predominancia del ruido (azul).	27
A.1. Distribución de <code>A_Zmass</code> por clase.	31
A.2. Distribución de <code>A_Dr_Z</code> por clase.	32
A.3. Distribución de <code>B_Dr_Z</code> por clase.	32
A.4. Distribución de <code>B_Zmass</code> por clase.	33
A.5. Distribución de <code>C_Dr_Z</code> por clase.	33
A.6. Distribución de <code>C_Zmass</code> por clase.	34
A.7. Distribución de <code>D_Dr_Z</code> por clase.	34
A.8. Distribución de <code>D_Zmass</code> por clase.	35
A.9. Distribución de <code>nElectron</code> por clase.	35
A.10. Distribución de <code>nMuon</code> por clase.	36

Índice de cuadros

3.1. Muestras Monte Carlo utilizadas para el entrenamiento y evaluación del AutoEncoder. Los procesos WZ, ZZ y DY sirven como procesos de ruido (Modelo Estándar) y la muestra Wprime como señal (anomalía).	15
3.2. Capas y activaciones del AutoEncoder	19

Capítulo 1

Introducción

1.1. Contexto

Desde el descubrimiento del bosón de Higgs en 2012, los experimentos del CERN (Centro Europeo para la Investigación Nuclear) se han enfocado en la búsqueda de nuevas partículas y fuerzas que puedan dar explicación a fenómenos más allá del modelo estándar, como por ejemplo la composición de la materia oscura. Poner a prueba los diferentes modelos de nueva física implica manejar enormes cantidades de datos, los cuales requieren técnicas avanzadas de programación para su análisis. En años recientes, los modelos de aprendizaje automático (machine learning) han permitido optimizar diversas tareas, como la simulación de colisiones, la identificación de patrones en grandes volúmenes de datos y la mejora en la reconstrucción de eventos. En particular, el uso de los llamados modelos auto codificadores (AutoEncoders) ha sido fundamental para la clasificación de eventos anómalos, una tarea de relevancia en la física de altas energías.

1.2. Antecedentes

Las colisiones de haces de protones del Gran Colisionador de Hadrones (LHC) son usadas para poner a prueba el modelo estándar de física de partículas, el cual abarca el entendimiento que tenemos hasta hoy de las partículas elementales y sus interacciones. Sin embargo, hay fenómenos observados que el modelo estándar no puede explicar, como la composición de la materia oscura, masas de neutrinos o la asimetría de materia-antimateria; aun así, en un sistema controlado de colisiones de partículas, las observaciones siguen funcionando de acuerdo con el modelo estándar [1].

Los datos producidos por colisiones de partículas son extremadamente grandes. Las técnicas de machine learning aplicadas en este contexto permitirían optimizar diversas tareas, como, por ejemplo, mejorar la eficiencia en la reconstrucción de eventos físicos, el tiempo de ejecución para simulaciones, el reconocimiento de patrones y las calibraciones, así como la implementación en tiempo real de algoritmos, entre otras [2].

La detección de anomalías (eventos o señales detectadas en experimentos que no concuerdan con los modelos teóricos que podrían ser indicativos de nuevos fenómenos físicos o simplemente ser errores experimentales o ruido) juega un papel importante en diversas disciplinas científicas, ayudando al descubrimiento de eventos poco probables que se desvían significativamente de lo esperado. Los enfoques tradicionales de detección de anomalías se ven limitados por la complejidad y variabilidad de los datos, lo que dificulta la definición de reglas que cubran todos los escenarios posibles. Como resultado, las técnicas de machine learning han ganado popularidad como una herramienta más flexible y poderosa [3]. Algunas de las técnicas se aprovechan principalmente de la simulación de Monte Carlo y se utiliza para comparar distribuciones en los datos observados con simulaciones a través de varios parámetros físicos (energía de las partículas, número de jets, rapidez, etc) e histogramas. El inconveniente de esta

metodología es que debido al gran tamaño del parámetro de espacio donde se busca la información, una observación que puede parecer estadísticamente significativa podría ser el resultado de una fluctuación estadística [4].

El experimento CMS (Solenoides compacto de muones) [5] ha estado recolectando información de gran cantidad de colisiones protón-protón producidas en el LHC por más de una década [6]. El detector CMS lee más información de la que puede ser procesada, reconstruida y analizada. Para poder usar alguna información de los Tb/s que se están generando, se necesita una reducción de más del 99 % [7].

Utilizando machine learning se han encontrado numerosas aplicaciones relevantes en física de partículas, donde muchas tareas requieren la clasificación de espacios variables en dimensiones superiores. Un reto de incorporar técnicas de machine learning en el análisis de datos de física de altas energías es que las herramientas son usualmente optimizadas para realizar una tarea específica que se desvía del objetivo de buscar una nueva partícula o probar nuevas teorías físicas. En los últimos años, los avances en machine learning han provocado el desarrollo de nuevas herramientas de análisis de datos en la física de altas energías. Las investigaciones en los campos de la estadística y machine learning ofrecen soluciones prometedoras para los retos de análisis de física de partículas. Un área con un potencial de mejora y relación con las colisiones de partículas es la detección de anomalías. En el contexto del CMS una anomalía puede ser por exceso de eventos (número alto de eventos en ciertas configuraciones de energía, masa invariante, o en la producción de partículas específicas), comportamientos no esperados en la producción de jets o incluso por eventos de partículas desconocidas, que hasta la fecha no se ha confirmado ninguna; técnicas recientes [8] han mejorado la comprensión de los datos enfocándose en la rapidez de detección de anomalías con datos del LHC de manera práctica [9].

1.3. Planteamiento del problema

Desde que se empezaron a hacer colisiones protón-protón en el LHC en el 2011, las colaboraciones del ATLAS (Aparato Toroidal del LHC) [10] y CMS han establecido límites estrictos para una variedad de modelos de nueva física, empujando el rango de masa permitido para nuevas partículas postuladas dentro de la escala de TeV. Es posible que estas partículas aún no se hayan observado porque son demasiado pesadas para ser producidas en el LHC, o tienen una sección transversal muy pequeña para ser detectadas con el tamaño de datos actual. También podría ser que nuevas partículas son cinemáticamente accesibles y se producen a velocidades observables, pero los métodos actuales de detección, que incluyen tanto los detectores como los enfoques estadísticos utilizados para analizar los datos, pueden limitar su descubrimiento. En este contexto, el aprendizaje automático puede ofrecer mejoras en la calibración de los detectores y en el análisis de datos, aumentando así la probabilidad de identificar partículas nuevas. La búsqueda de nuevos procesos físicos en colisionadores de partículas se realiza normalmente a ciegas o siguiendo procesos que podrían estar ocultando la verdadera nueva física. Se han realizado cientos de búsquedas para potenciales nuevas partículas diferentes, pero hasta ahora ninguna ha sido descubierta [4]. Esto ha provocado un mayor interés en estrategias de búsqueda de modelos independientes como, por ejemplo, usando modelos AutoEncoders. Estas búsquedas no son nuevas en física de alta energía, sin embargo, se siguen buscando estrategias que permitan minimizar el uso de hipótesis de posibles señales.

1.4. Solución propuesta

Se usará información de procesos normales (aquellos que se ajustan a las predicciones y que han sido ampliamente observados y estudiados en experimentos) del modelo estándar para el entrenamiento de un modelo autoencoder y clasificar eventos anómalos, probar su desempeño con datos experimentales e identificar eventos que pudieran ser indicativos de nueva física, separar esos datos anómalos y estudiarlos por medio de visualización y paquetería de análisis, buscando dar una interpretación física o experimental a la clasificación dentro del grupo anómalo. Adicionalmente se buscará comparar este modelo con otros alternativos como los algoritmos de clustering y medir su desempeño.

1.5. Objetivo general

Desarrollar un modelo de red neuronal AutoEncoder para la detección de eventos anómalos usando como datos procesos característicos de la física de altas energías

1.6. Objetivos específicos

- Identificar elementos básicos de un modelo de red neuronal AutoEncoder (AE) por medio de revisión de literatura y ejemplos disponibles
- Analizar muestras simuladas de procesos de física de altas energías.
- Visualizar los eventos anómalos seleccionados por el modelo AE.
- Comparar los resultados obtenidos por medio del modelo AE con respecto a otros métodos alternos (ej. clustering)

1.7. Justificación

En este momento, la física de altas energías se encuentra en una etapa de incertidumbre, ya que no se sabe exactamente qué modelo utilizar para la búsqueda de nueva física. Si bien existen varios modelos que postulan la existencia de nuevas partículas y fuerzas, muchos de estos han sido desfavorecidos por las mediciones experimentales hasta el día de hoy, ya que no cumplen con las predicciones verificables en los resultados de las colisiones. En este contexto, los modelos de aprendizaje automático pueden proporcionar herramientas adicionales para explorar los datos, permitiendo identificar señales que no se han observado con técnicas tradicionales y que podrían validar alguno de los modelos aún no descartados o descubrir nuevas señales que apunten a la presencia de nueva física.

1.8. Impactos

- Contribución al avance del análisis de datos en física de partículas: Los resultados obtenidos podrían mejorar los métodos de detección de eventos anómalos, facilitando el desarrollo de nuevas técnicas para la búsqueda de fenómenos físicos más allá del Modelo Estándar, lo que aportará al avance del conocimiento científico y al fortalecimiento de investigaciones relacionadas.
- Impacto en la formación académica y profesional: La experiencia adquirida en el manejo de grandes volúmenes de datos y en el entrenamiento de modelos AutoEncoders contribuirá a la formación profesional especializada en análisis de datos.
- Se puede usar una metodología similar para detectar eventos raros de neutrinos en medio de grandes volúmenes de ruido de fondo. Adaptando el AutoEncoder para identificar interacciones anómalas de neutrinos de alta energía, se podría mejorar la detección y reducir falsos positivos. Esto permitiría avanzar en la comprensión de fenómenos cósmicos extremos, como supernovas, y mejorar la precisión en el análisis de eventos raros en tiempo real.

1.9. Alcances y delimitaciones

1.9.1. Alcances

- El modelo se entrenará con una variedad de muestras que cubren los rangos del Modelo Estándar. Su capacidad para generalizar estará limitada por el contexto específico de estados finales y topologías consideradas.

- El estudio se centrará en eventos relacionados con la detección de anomalías en el contexto del experimento CMS del CERN, proporcionando una base para investigaciones futuras en la búsqueda de nueva física.

1.9.2. Delimitaciones

- La validez del modelo estará restringida al conjunto de datos utilizado y a la topología específica del estudio. El enfoque se centrará en una prueba de concepto para caracterizar eventos de interés, sirviendo como base para desarrollos futuros más detallados.
- El análisis se limitará a los datos disponibles del experimento CMS del CERN, sin considerar otros posibles conjuntos de datos o experimentos que podrían influir en los resultados o en la generalización del modelo.

Capítulo 2

Marco Teórico

2.1. Introducción

En este capítulo se proporciona una revisión de los conceptos fundamentales en el desarrollo de este proyecto. Primero se aborda un breve contexto de la física de altas energías, incluyendo una descripción general del Modelo Estándar y la búsqueda de nueva física. Después, el aprendizaje automático y sus aplicaciones en la física de altas energías, con especial énfasis en el uso de redes neuronales autoencoders. Finalmente, se discuten los métodos basados en deep learning aplicados a la detección de anomalías en colisiones de altas energías, se destaca el papel de los autoencoders como una herramienta clave en la detección de eventos poco comunes que podrían indicar la presencia de nueva física.

2.2. Física de altas energías

La física de altas energías (por sus siglas en inglés HEP) explora los bloques fundamentales del universo y sus interacciones. El Modelo Estándar es una teoría central en este campo, describiendo las partículas elementales conocidas y tres de las cuatro fuerzas fundamentales: la fuerza electromagnética, la fuerza débil y la fuerza fuerte [11], [12].

2.2.1. El Modelo Estándar

El Modelo Estándar se compone de dos tipos principales de partículas: fermiones y bosones, tal como se muestra en el diagrama de la Figura 2.1. Los fermiones, que incluyen quarks y leptones, constituyen la materia. Los quarks forman partículas compuestas como protones y neutrones, mientras que los leptones incluyen electrones, muones, taus y sus correspondientes neutrinos. Los bosones, por otro lado, son responsables de mediar las interacciones entre las partículas [12]. El Modelo Estándar ha sido validado por numerosos experimentos, mostrando una precisión asombrosa en la predicción de diversos fenómenos. Los experimentos en colisionadores de partículas, como el LHC, han proporcionado una gran cantidad de datos que confirman las predicciones del Modelo Estándar. Por ejemplo, las mediciones de parámetros electrodébiles realizadas en el colisionador de electrones-positrones LEP del CERN y en el experimento SLD, instalado en el colisionador lineal SLC del SLAC, muestran un acuerdo impresionante con las predicciones del Modelo Estándar.[11], [12].

A pesar de su éxito, el Modelo Estándar no es una teoría completa. Hay fenómenos que no puede explicar, lo que sugiere la existencia de "nueva física" más allá de sus límites. Algunas de las limitaciones más notables son:

- La exclusión de la gravedad: el Modelo Estándar no incorpora la gravedad, la cuarta fuerza fundamental. Unificar la gravedad con las otras fuerzas en un marco teórico único es uno de los mayores desafíos de la física moderna [9].

Standard Model of Elementary Particles

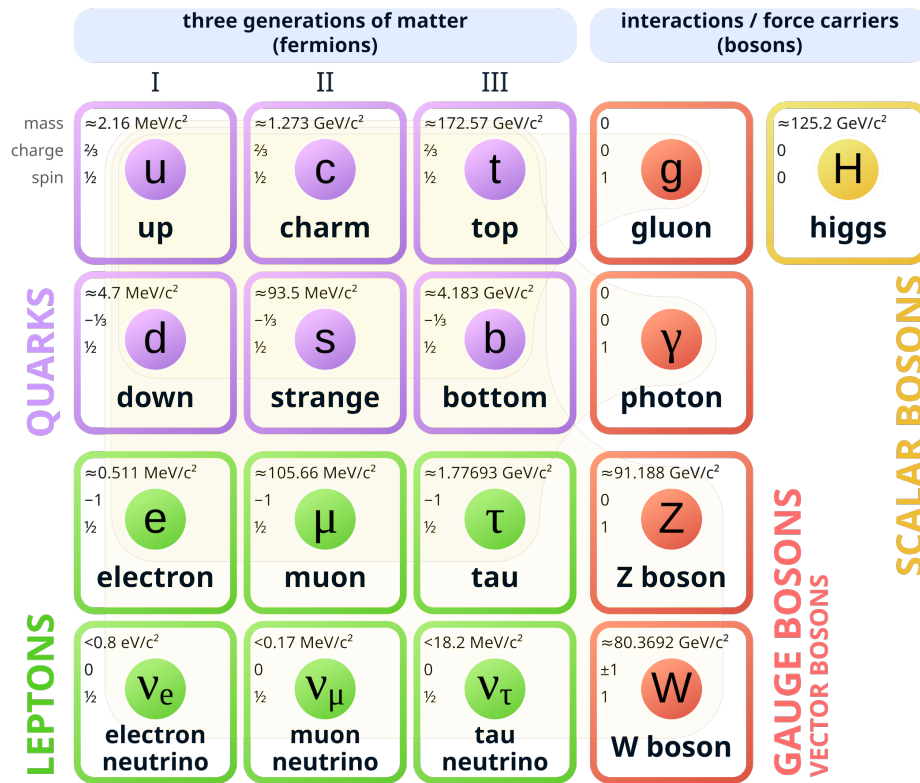


Figura 2.1: Diagrama simplificado del Modelo Estándar de física de partículas. Se muestran los seis quarks (*up*, *down*, *charm*, *strange*, *top*, *bottom*), los seis leptones (electrón, muón, tau y sus neutrinos), los bosones gauge (γ , W^\pm , Z^0 , gluones) y el bosón de Higgs (H^0) [13].

- El enigma de la materia oscura y la energía oscura: la materia oscura y la energía oscura, que constituyen la mayor parte del contenido energético del universo, no tienen explicación dentro del Modelo Estándar. Su naturaleza y propiedades son un misterio que requiere nueva física para ser resuelto [9].
- La masa de los neutrinos: el Modelo Estándar inicialmente predecía neutrinos sin masa. Sin embargo, experimentos han demostrado que los neutrinos oscilan entre diferentes tipos, lo que implica que deben tener masa. Explicar el origen y la magnitud de la masa de los neutrinos requiere extender el Modelo Estándar [12].
- La asimetría materia-antimateria: el Modelo Estándar no puede explicar por qué el universo observable está dominado por materia y no por antimateria. Esta asimetría es un enigma fundamental que requiere una comprensión más profunda de las leyes de la física [12].

Para abordar las limitaciones del Modelo Estándar, los físicos de altas energías están constantemente buscando nuevas partículas e interacciones. Experimentos de alta energía, como los realizados en el LHC, son esenciales en esta búsqueda. Los físicos analizan los datos de estas colisiones en busca de señales que no se ajusten al Modelo Estándar, lo que podría indicar la presencia de nuevas partículas o fuerzas [11].

Las técnicas de aprendizaje automático están desempeñando un papel cada vez más importante en la búsqueda de nueva física. Los algoritmos de ML se utilizan para analizar grandes conjuntos de datos de colisiones, buscando patrones y anomalías que podrían ser señales de nueva física. Por ejemplo, los autoencoders (AE) y los variational autoencoders (VAE) se utilizan para detectar eventos atípicos en los datos que podrían indicar la presencia de nueva física [4].

2.2.2. Búsqueda física mas allá del Modelo Estándar

Los procesos de interés en el CMS son aquellos que pueden revelar nueva física más allá del Modelo Estándar. Estos procesos suelen involucrar la producción de partículas pesadas o interacciones inusuales. Algunos ejemplos son:

- Producción del bosón de Higgs: El descubrimiento del bosón de Higgs fue un hito para el CMS y la física de partículas en general. El estudio de sus propiedades es crucial para comprender el mecanismo de Higgs, responsable de la masa de las partículas elementales [6].
- Producción de quarks top: El quark top es la partícula elemental más pesada conocida. Su estudio proporciona información valiosa sobre las interacciones fuertes y puede revelar nueva física [14].
- Eventos con alto momento transversal (p_T): Cantidad de movimiento de una partícula en la dirección perpendicular al eje de la aceleración o colisión. Pueden indicar la producción de partículas pesadas o interacciones a altas energías. Estos eventos son particularmente interesantes en la búsqueda de nueva física [1].
- Alto momento transversal faltante (MET_{pt}): Una gran energía no balanceada en el plano transversal indica la presencia de partículas neutras no detectadas (neutrinos o posibles candidatos de materia oscura). Los eventos con MET elevado son, por tanto, firmas clave en análisis de procesos exóticos invisibles [15].

En el contexto de las colisiones del CMS, las anomalías son eventos que se desvían significativamente de las predicciones del Modelo Estándar. Estas anomalías pueden ser indicios de nueva física o problemas en la comprensión del detector o del análisis de datos [4].

Dos tipos principales de anomalías son:

- Detección de valores atípicos: Se centra en identificar eventos inusuales o inesperados que se destacan de la norma. Estos valores atípicos se encuentran típicamente en las colas de las distribuciones, lo que representa eventos raros [4].

- Detección de sobredensidades: Esta detección se asemeja a la búsqueda tradicional de "bumps" (protuberancias), donde se buscan excesos localizados de puntos de datos en comparación con la distribución esperada. Estas sobredensidades o resonancias pueden indicar la presencia de nuevas partículas o procesos físicos inesperados [4].

El CMS produce una cantidad enorme de datos, mucho más de lo que se puede almacenar o analizar. Los "triggers" son sistemas de selección de eventos en tiempo real que deciden qué eventos se guardan para su posterior análisis y cuáles se descartan [4].

- Trigger de Nivel 1 (L1): Es un sistema basado en hardware que utiliza información rápida de los detectores para seleccionar eventos potencialmente interesantes en tiempo real. El L1 reduce la tasa de eventos de 40 MHz a 100 kHz [4], [7].
- Trigger de Alto Nivel (HLT): Es un sistema basado en software que utiliza algoritmos más sofisticados para analizar los eventos seleccionados por el L1. El HLT reduce aún más la tasa de eventos a 1 kHz [4].

Los "triggers" tradicionales del CMS se basan en criterios cinemáticos, como la presencia de partículas con alto p_T (momento transversal: el componente del momento de la partícula perpendicular al eje del haz). Sin embargo, estos "triggers" pueden tener una sensibilidad reducida a eventos con nueva física que no producen partículas de alta energía [4].

El presente estudio se enfocará en la caracterización de eventos con al menos 3 leptones, entendiéndose leptones como electrones (e) o muones (μ), los taus no se consideran por la complejidad experimental de identificarlos correctamente. En este supuesto, la selección de eventos se puede dividir en tres canales de producción. Canal A caracterizado por la presencia de al menos 3 electrones (eee), canal B caracterizado por la presencia de al menos 3 electrones y un muón ($ee\mu$) canal C caracterizado por la presencia de al menos 2 muones y un electrón ($\mu\mu e$) y canal D caracterizado por la presencia de al menos 3 muones ($\mu\mu\mu$). En el modelo estándar, eventos de este tipo se pueden producir por diversos mecanismos; los más comunes son el decaimiento de pares de bosones (WZ , ZZ) o decaimiento de pares de quarks top ($t\bar{t}$) los cuales pueden producir eventos con 3 leptones y neutrinos (los cuales escapan a la detección). Por lo que se usarán este tipo de muestras para caracterizar al Modelo Estándar, es decir, los datos esperados (normales). Por otro lado, un modelo de nueva física que puede producir la misma topología (3 leptones) es la producción del hipotético bosón W-prime (W') [16] que sería una partícula parecida al bosón W pero con un valor de masa mucho mayor. Esta partícula, en principio, podría decaer a un par de bosones (WZ) y después producir una señal con 3 leptones y un neutrino, por lo que este proceso se utilizará para caracterizar a los eventos de señal (anómalos). Una representación de la topología de estudio se presenta en el diagrama de Feynman de la Figura 2.2.

2.3. Machine learning

El aprendizaje automático (ML), también conocido como aprendizaje de máquinas, es un campo que permite que las computadoras hagan predicciones exitosas utilizando datos previos como referencia para identificar patrones y tendencias. En lugar de especificar manualmente un conjunto complejo de reglas, los métodos de aprendizaje automático pueden construir automáticamente un modelo estadístico a partir de datos. Estos modelos pueden ser entrenados usando grandes muestras de datos ya que el proceso de entrenamiento es automatizado [17]. Las cantidades observables que se ingresan en un algoritmo de aprendizaje automático se llaman 'características'. El algoritmo aprende una asignación de estas características a las variables de salida deseadas ajustando los parámetros del modelo utilizando los datos de entrenamiento disponibles. Es crucial que las características sean relevantes para la predicción de las salidas [17].

En el aprendizaje supervisado, se proporciona al algoritmo un conjunto de datos de entrenamiento que incluye las entradas y las salidas deseadas. El objetivo del algoritmo es aprender una función que

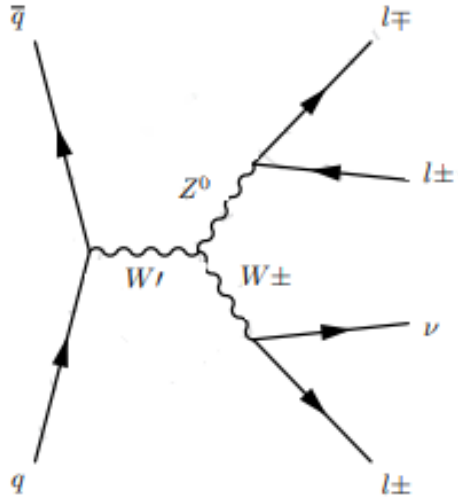


Figura 2.2: Diagrama de Feynman de la topología de estudio, en este caso representando la producción de una partícula W , su decaimiento a WZ y finalmente 3 leptones.

pueda mapear las entradas a las salidas con la mayor precisión posible. Este tipo de aprendizaje se puede dividir en dos categorías: clasificación y regresión. La clasificación se ocupa de predecir una salida discreta, como si un correo electrónico es spam o no, o si una imagen contiene un gato o un perro. La regresión, por otro lado, se ocupa de predecir una salida continua, como el precio de una casa o la temperatura de una ciudad. En el aprendizaje no supervisado, el algoritmo solo recibe las entradas y debe encontrar patrones o estructuras en los datos por sí mismo. Un ejemplo de aprendizaje no supervisado puede ser la agrupación en clústeres, donde el algoritmo debe agrupar puntos de datos similares. [17]

Algunas consideraciones que se tienen que tener en cuenta son [17]:

- Evaluación del Modelo: El diseño del experimento para probar la capacidad de generalización de un modelo debe realizarse correctamente, considerando aspectos importantes como la elección de las muestras de entrenamiento y la aleatorización.
- Complejidad del Modelo: Elegir la complejidad adecuada del modelo es crucial para el éxito de un experimento de aprendizaje automático.
- Métricas de Rendimiento: La selección de métricas de rendimiento es fundamental para evaluar el éxito de un experimento de aprendizaje automático.

En el contexto de física de partículas el aprendizaje automático ha desempeñado un papel crucial en el análisis de datos de física de altas energías durante décadas. En 2012 permitió el desarrollo de herramientas que podían manejar problemas de mayor dimensión y más complejos que antes [9]. El ML se utiliza en la física de altas energías para una variedad de tareas, que incluyen [2]:

- Reconstrucción de trazas
- Identificación de objetos
- Clasificación de eventos
- **Detección de anomalías**
- Optimización de recursos informáticos

2.3.1. Redes neuronales

Una red neuronal es un modelo computacional inspirado en la estructura y el funcionamiento del cerebro humano. Consiste en un conjunto de nodos interconectados, también conocidos como neuronas artificiales, que procesan y transmiten información [18], [19].

Cada neurona artificial recibe una serie de entradas (inputs) y realiza una operación matemática sobre ellas, como una suma ponderada. El resultado de esta operación se pasa a través de una función de activación, que introduce no linealidad en el modelo. La salida de una neurona puede ser la entrada de otras neuronas, creando una red de procesamiento [18], [19]. Las redes neuronales se utilizan para una gran variedad de tareas, como:

- Clasificación: Asignar una entrada a una categoría específica (por ejemplo, identificar una imagen como un gato o un perro) [19].
- Regresión: Predecir un valor continuo (por ejemplo, predecir el precio de una acción) [19].
- Aprendizaje no supervisado: Descubrir patrones en los datos sin tener etiquetas predefinidas (por ejemplo, agrupar clientes en diferentes segmentos) [19].

La capacidad de procesamiento de una red neuronal reside en las fuerzas de conexión entre las neuronas, conocidas como pesos [18]. Estos pesos se ajustan durante el proceso de aprendizaje, en el que la red se expone a un conjunto de datos de entrenamiento y se le indica la salida deseada para cada entrada. A través del aprendizaje, la red ajusta sus pesos para minimizar la diferencia entre sus predicciones y las salidas deseadas [19]. Existen diferentes tipos de redes neuronales, como:

- Redes neuronales feedforward (FNN): La información fluye en una sola dirección, desde la capa de entrada hasta la capa de salida [20].
- Redes neuronales recurrentes (RNN): La información puede fluir en ambas direcciones, lo que les permite procesar secuencias de datos, como el lenguaje natural [21].
- Las redes neuronales convolucionales (CNN): Son un caso especial de red neuronal artificial (ANN) que sobresale en el procesamiento de imágenes al preservar las relaciones espaciales entre los píxeles. A diferencia de las ANN simples que tratan cada píxel individualmente, las CNN procesan parches de la imagen, manteniendo el contexto espacial de las características extraídas [19], [20].

Deep learning

El deep learning, también conocido como aprendizaje profundo, se puede entender como una subcategoría de ML que se centra en el uso de redes neuronales con múltiples capas. A diferencia de las redes neuronales tradicionales, que suelen tener solo unas pocas capas, las redes de deep learning pueden tener decenas o incluso cientos de capas. Esta profundidad les permite aprender características más abstractas y complejas de los datos, lo que las hace más potentes y versátiles. Las redes de deep learning están compuestas por módulos individuales que se pueden combinar de diferentes maneras. Esto permite a los investigadores experimentar con diferentes arquitecturas de red y encontrar la mejor para una tarea en particular. Además, los componentes de la red son diferenciables, lo que significa que se pueden entrenar utilizando algoritmos de optimización basados en gradientes. Un ejemplo de la aplicación de deep learning en la física de colisionadores es la clasificación de jets, donde se utilizan redes neuronales profundas para identificar la presencia de ciertas partículas en los eventos de colisión [9].

2.3.2. Autoencoders

Los autoencoders son un tipo de algoritmo cuyo objetivo principal es aprender una representación 'informativa' de los datos, que puede utilizarse para diferentes aplicaciones, aprendiendo a reconstruir un conjunto de observaciones de entrada lo suficientemente bien. En otras palabras, un autoencoder busca aprender la función de identidad de manera aproximada [22].

Las partes principales de un autoencoder son tres [22]:

- **Codificador:** El codificador es una función que toma las observaciones de entrada y las normalmente comprime en una representación de menor dimensionalidad llamada representación de características latentes.
- **Representación de características latentes:** Es un tensor de números reales que representa la información esencial de las observaciones de entrada de forma comprimida.
- **Decodificador:** El decodificador es una función que toma la representación de características latentes y la utiliza para reconstruir las observaciones de entrada originales.

El codificador es una función $g(\mathbf{x}_i)$ que mapea las observaciones de entrada \mathbf{x}_i a la representación de características latentes $\mathbf{h}_i = g(\mathbf{x}_i)$. La función $g(\mathbf{x}_i)$ puede ser una red neuronal, una función lineal u otra función. El decodificador es una función $f(\mathbf{h}_i)$ que mapea la representación de características latentes \mathbf{h}_i a las observaciones reconstruidas $\tilde{x}_i = f(\mathbf{h}_i) = f(g(\mathbf{x}_i))$ [1]. La función $f(g(\mathbf{x}_i))$ también puede ser una red neuronal, una función lineal u otra función.

El entrenamiento de un autoencoder consiste en encontrar las funciones $g(\mathbf{x}_i)$ y $f(g(\mathbf{x}_i))$ que minimicen la función de pérdida [1]. Esto se puede expresar matemáticamente como [22]:

$$\operatorname{argmin}_{f,g} < [\Delta(\mathbf{x}_i, f(g(\mathbf{x}_i)))] > \quad (2.1)$$

donde:

Lo que hay dentro de $\Delta()$ es la función de pérdida, que mide la diferencia entre las observaciones de entrada y las observaciones reconstruidas. $< >$ indica el promedio sobre todas las observaciones.

Las funciones de pérdida más comunes utilizadas para entrenar autoencoders son:

- **MSE (Error Cuadrático Medio):** El MSE calcula el promedio de los cuadrados de las diferencias entre las salidas y los valores objetivo. Es una función de pérdida común para problemas de regresión, y también se puede usar en autoencoders para medir la calidad de la reconstrucción de la entrada [22].

$$L_{MSE} = \text{MSE} = \frac{1}{M} \sum_{i=1}^M |\mathbf{x}_i - \tilde{\mathbf{x}}_i|^2 \quad (2.2)$$

- **BCE (Entropía Cruzada Binaria):** La BCE es una función de pérdida utilizada en problemas de clasificación binaria. Se basa en la probabilidad de que la salida de la red neuronal prediga correctamente la clase del valor objetivo. Es una función de pérdida común para autoencoders cuando los valores de entrada son binarios [22].

$$L_{CE} = -\frac{1}{M} \sum_{i=1}^M [\mathbf{x}_i \log(\tilde{\mathbf{x}}_i) + (1 - \mathbf{x}_i) \log(1 - \tilde{\mathbf{x}}_i)] \quad (2.3)$$

En la mayoría de las arquitecturas, el codificador y el decodificador son redes neuronales que se entrenan utilizando retropropagación con bibliotecas de software como TensorFlow o PyTorch [22].

La función de activación de la capa de salida del autoencoder juega un papel importante. Entre las más utilizadas tenemos ReLU, sigmoid y SeLU.

- **ReLU (Unidad Lineal Rectificada):** La función ReLU es una función simple y efectiva que se define como el máximo entre 0 y el valor de entrada. Esta función es popular debido a su simplicidad y a que ayuda a evitar problemas como la desaparición del gradiente. Es una buena opción cuando los valores de entrada abarcan un rango amplio de valores positivos. En cambio, si la entrada puede tener valores negativos, ReLU no sería una buena opción y se recomienda usar la función identidad [22].

$$\text{ReLU}(x) = \max(0, x) \quad (2.4)$$

- **Sigmoide:** La función sigmoide es una función en forma de S que mapea la entrada a un rango entre 0 y 1. Es útil para problemas de clasificación binaria donde la salida representa la probabilidad de pertenecer a una clase. La función sigmoide se utiliza a menudo en la capa de salida de los autoencoders, especialmente cuando los valores de entrada están normalizados entre 0 y 1 [22].

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (2.5)$$

- **SeLU (Scaled Exponential Linear Unit):** La función SeLU es una variante de normalización autónoma que, en combinación con una correcta inicialización de pesos, mantiene de forma automática la media cercana a 0 y la varianza cercana a 1 a lo largo de capas profundas. Esta propiedad elimina la necesidad de técnicas como *batch-norm* y acelera la convergencia en redes densas profundas [23].

$$\text{SeLU}(x) = \lambda \begin{cases} x & \text{si } x > 0, \\ \alpha e^x - \alpha & \text{si } x \leq 0, \end{cases} \quad (2.6)$$

donde $\lambda \approx 1,0507$ y $\alpha \approx 1,6733$.

El error de reconstrucción (RE) es una métrica que indica qué tan bien (o mal) el autoencoder pudo reconstruir la observación de entrada \mathbf{x}_i . El RE más común es el MSE (ecuación 2.2).

El RE se utiliza a menudo cuando se realiza la detección de anomalías con autoencoders, ya que un RE alto indica que el autoencoder no pudo reconstruir bien la entrada, lo que sugiere que la entrada puede ser anómala [22].

Para evitar que el autoencoder aprenda la función de identidad y para fomentar la extracción de características significativas, se utilizan técnicas de regularización [22].

- **Bottleneck :** reducir la dimensionalidad de las características latentes, obligando al autoencoder a aprender una representación comprimida de los datos.
- **Regularización L1 o L2:** agregar un término de regularización a la función de pérdida para fomentar la escasez en la representación de características latentes.
- **Vincular los pesos:** hacer que los pesos del codificador sean iguales a los pesos del decodificador.

Los autoencoders pueden ser propensos al sobreajuste y a la inestabilidad de los resultados, especialmente en el caso de la detección de anomalías. Es importante ser consciente de estas limitaciones y validar los resultados con diferentes modelos [22].

Algunos tipos de autoencoders son:

- **Autoencoders de alimentación directa (FFA):** Es un tipo de red neuronal feedforward (FNN) diseñada específicamente para realizar tareas de codificación y decodificación, como la reducción de dimensionalidad o la reconstrucción de datos.
- **Autoencoders convolucionales:** Utilizan capas convolucionales para el codificador y el decodificador, lo que los hace más adecuados para datos de imágenes.
- **Autoencoders de eliminación de ruido:** Se entrenan con datos ruidosos para aprender a reconstruir los datos originales sin ruido.

2.3.3. Métricas de validación de anomalías

En la búsqueda de anomalías mediante aprendizaje automático, se emplean métricas de evaluación similares a las de problemas de clasificación binaria para cuantificar el desempeño. Una de las herramientas principales es la curva ROC (Receiver Operating Characteristic), que muestra la tasa de verdaderos positivos frente a la de falsos positivos al variar el umbral de decisión del modelo [24]. El resumen numérico más utilizado de esta curva es el área bajo la curva (AUC), que proporciona una medida única de la capacidad de discriminación del modelo independiente de un umbral específico [24]. Un AUC cercano a 1 indica que el modelo asigna sistemáticamente puntuaciones de anomalía más altas a los eventos señal que a los de fondo. En aplicaciones de detección de anomalías en física de partículas con autoencoders u otros modelos no supervisados, es común reportar tanto la curva ROC como el valor de AUC para distintos tipos de señales simuladas [25], [26]. Estas métricas permiten comparar objetivamente modelos y ajustar hiperparámetros, y han sido empleadas, por ejemplo, para evaluar autoencoders en la identificación de jets inusuales o eventos raros del LHC [25]. Adicionalmente, en ausencia de una etiqueta clara de señal, a veces se define un umbral de anomalía basado en percentiles del puntaje: por ejemplo, marcar como anomalías el X % de eventos con mayor error de reconstrucción. Este criterio por percentil fija la tasa de falsos positivos (identificando qué fracción de datos de fondo serían considerados anómalos) y sirve para definir regiones de interés que luego pueden examinarse más detalladamente.

Más allá de las métricas numéricas, es valioso interpretar la respuesta del modelo explorando la estructura interna de los datos en espacios de menor dimensión. Para ello se utilizan técnicas de visualización del espacio latente. Un método lineal clásico es el análisis de componentes principales (PCA), que proyecta los datos en las direcciones de mayor varianza capturando patrones globales [24]. En el contexto de autoencoders, aplicar PCA sobre las representaciones latentes puede revelar si los eventos anómalos se separan de los de fondo en las primeras componentes principales. Asimismo, técnicas no lineales como t-SNE (t-distributed Stochastic Neighbor Embedding) permiten mapear datos de alta dimensionalidad a dos dimensiones preservando relaciones de proximidad [27]. Esta técnica es particularmente útil para visualizar cómo el autoencoder organiza los eventos en el espacio latente: por ejemplo, se pueden proyectar los vectores latentes de eventos reconstruidos y ver si forman cúmulos distinguibles. Estudios recientes de aprendizaje no supervisado en HEP han empleado t-SNE para inspeccionar la separación entre eventos simulados de nueva física y eventos de fondo en el espacio latente del modelo [26]. Estas visualizaciones complementan las métricas cuantitativas, proporcionando intuición sobre la estructura de los datos y ayudando a comprender qué aprende el modelo al distinguir anomalías.

2.4. Deep learning aplicado a física de altas energías

El deep learning es una herramienta poderosa en la física de altas energías, especialmente en la búsqueda de nueva física más allá del Modelo Estándar. Los grandes conjuntos de datos producidos por colisionadores como el CMS y la complejidad de los procesos físicos involucrados hacen que las técnicas de ML sean ideales para analizar estos datos y extraer información relevante [9].

Los autoencoders se utilizan a menudo para detectar anomalías en conjuntos de datos de física de altas energías. La idea básica es entrenar un autoencoder en eventos del Modelo Estándar (eventos "normales"). Una vez que el autoencoder está entrenado, se utiliza para reconstruir eventos nuevos. Los eventos que el autoencoder no puede reconstruir con precisión se consideran anómalos, ya que sus características difieren de los eventos en los que fue entrenado [4].

Un ejemplo de la aplicación de autoencoders para la detección de anomalías en el CMS se presenta en [6]. En este trabajo, se utiliza un autoencoder para monitorear la calidad de los datos del calorímetro electromagnético (ECAL). El autoencoder se entrena en histogramas de ocupación del ECAL procesados como imágenes, y se utiliza para identificar anomalías en tiempo real en los datos de las colisiones del LHC [6].

2.4.1. Preprocesamiento de datos en HEP

En los experimentos de física de altas energías, los datos se almacenan en formatos estructurados para facilitar su manejo. Un formato ampliamente utilizado es ROOT, un formato binario y framework orientado a objetos desarrollado en CERN para almacenar eventos y sus múltiples variables asociadas [28]. Por ejemplo, el experimento CMS organiza sus datos en varios niveles: tras el reconocimiento inicial se produce el formato AOD (Analysis Object Data), del cual se derivó un formato más compacto llamado Mini-AOD introducido en Run 2 [29]. El Mini-AOD reduce el tamaño de los eventos a aproximadamente un 10 % del AOD original mediante la eliminación de información redundante y la reducción de la precisión donde no es necesaria, manteniendo suficiente detalle para la mayor parte de los análisis de física [29]. Más recientemente, CMS implementó NanoAOD, un formato aún más ligero enfocado en análisis que reduce aún más el volumen de datos por evento [30]. NanoAOD permite distribuir conjuntos de datos de análisis de forma eficiente, conteniendo sólo las variables más esenciales por evento [30].

Durante el flujo de análisis, es habitual aplicar un filtrado de eventos (*skim*) para seleccionar sólo aquellos eventos de interés (por ejemplo, los que pasan ciertos cortes o contienen objetos particulares), produciendo n-tuplas personalizadas más manejables. Este preprocesamiento reduce la cantidad de datos a procesar sin perder sensibilidad al fenómeno bajo estudio. Para integrar los datos de HEP con herramientas modernas de aprendizaje automático, a menudo se convierten las n-tuplas resultantes a formatos más accesibles como HDF5. Este formato jerárquico permite almacenar matrices numéricas de forma eficiente y es compatible con librerías de Python usadas en aprendizaje automático, facilitando la carga masiva de eventos en redes neuronales. Varios trabajos técnicos recientes de CMS destacan la importancia de traducir los datos de ROOT a formatos optimizados para análisis con ML, manteniendo la fidelidad de la información física mientras se mejora el rendimiento en la lectura/escritura de datos [28], [30].

2.4.2. Comparación con métodos clásicos

Históricamente, la búsqueda de nueva física en HEP se ha basado en métodos estadísticos clásicos que requieren una fuerte implicación manual y suposiciones a priori sobre la naturaleza de la señal. Un ejemplo prominente es el *bump hunting*, que consiste en buscar “bumps” o excesos localizados en distribuciones invariantes como la masa invariante de dos objetos, sobre un fondo esperado suave [31]. Esta técnica se implementa construyendo histogramas de eventos y ajustando el fondo con una función suave o con simulaciones de Monte Carlo, para luego identificar desviaciones significativas que puedan indicar una resonancia nueva.

Si bien los métodos clásicos como el bump hunting y los análisis basados en histogramas han sido exitosos en descubrimientos previos, presentan limitaciones importantes en la era de grandes volúmenes de datos y alta dimensionalidad del LHC. En primer lugar, suelen estar restringidos a unas pocas variables escalares definidas previamente (por intuición física, como masas o ángulos característicos), lo que puede dejar escapar señales cuya manifestación no sea un pico pronunciado en una distribución simple [31]. Los análisis con aprendizaje automático abordan esta limitación explorando espacios multidimensionales de observables de manera conjunta, pudiendo descubrir correlaciones sutiles que un análisis univariado no captaría. Además, los métodos tradicionales dependen fuertemente de simulaciones de Monte Carlo para modelar el fondo; si el modelo de fondo es inexacto o incompleto, la sensibilidad al descubrir una señal real se ve comprometida. En cambio, los enfoques modernos de anomalías con ML pueden entrenarse directamente con datos reales de fondo (en un esquema no supervisado o débilmente supervisado), reduciendo la dependencia en las simulaciones [31]. Finalmente, la necesidad de correcciones por efectos de exploración múltiple crece con el número de distribuciones examinadas en paralelo en un análisis clásico [32]. Los algoritmos basados en ML, al englobar muchas variables a la vez, proporcionan una forma más natural de escanear el espacio de datos completos sin incurrir en un peaje estadístico tan severo por pruebas múltiples, aunque requieren validaciones rigurosas para evitar sesgos.

Capítulo 3

Metodología

3.1. Introducción

La estrategia del análisis se puede dividir en los siguientes pasos, la selección de muestras de los procesos de estudio, la preparación de los datos y selección de variables a utilizar para el entrenamiento de la red autoencoder, el entrenamiento del modelo, evaluación del modelo y discusión de la validez del mismo. A continuación se detalla cada uno de los pasos.

3.2. Selección de muestras

El punto de partida del estudio es la selección de muestras las cuales son generadas por el método de simulación por Monte Carlo. Como es común en la física de partículas experimental las muestras se procesan de manera independiente para cada proceso de interés, el formato usado se conoce como **NanoAOD** el cual es un formato propio de la colaboración CMS y que es ampliamente usado por la colaboración ya que contiene la gran mayoría de las variables reconstruidas de las partículas que se utilizan en los análisis del CMS.

En la Tabla 3.1 se presenta una descripción de las muestras utilizadas en este estudio, indicando cuales se consideran muestras de ruido (SM) o de nueva física (Señal), además de una descripción breve del proceso físico en cuestión:

Etiqueta	Tipo	Descripción física
WprimeToWZToWlepZlep_M2000	Señal	Producción de un bosón W' (2 TeV) $\rightarrow WZ \rightarrow \ell\nu\ell\ell$, ejemplo canónico de resonancia pesada que extiende el Modelo Estándar.
WZto3LNu	SM	Producción electrodébil $WZ \rightarrow 3\ell + \nu$, referencia para topologías trileptónicas.
ZZto4L	SM	Producción $ZZ \rightarrow 4\ell$, topología limpia de cuatro leptones.
DYJetsToLL	SM	Drell-Yan $Z/\gamma^* \rightarrow \ell^+\ell^-$ con jets; controla colas de masa invariante.

Tabla 3.1: Muestras Monte Carlo utilizadas para el entrenamiento y evaluación del AutoEncoder. Los procesos WZ, ZZ y DY sirven como procesos de ruido (Modelo Estándar) y la muestra Wprime como señal (anomalía).

3.3. Procesamiento de Muestras

3.3.1. Reducción de muestras (skimming) y selección de variables

Las muestras originales (nanoAOD) pasan por un proceso de filtrado (skimming) por el cual se reduce el número de variables originales y se crean nuevas variables relacionadas con el proceso de estudio. En el caso de este estudio se consideran eventos con al menos 3 leptones, además del neutrino que no se detecta de manera directa pero que su presencia se infiere por medio de la energía faltante (missing energy) en el evento. A continuación se presentan algunas de las variables que se guardan en las muestras filtradas:

- A_Zmass: Masa invariante del par de leptones en el canal A ($e e e$).
- B_Zmass: Masa invariante del par de leptones en el canal B ($e e \mu$).
- C_Zmass: Masa invariante del par de leptones en el canal C ($e \mu \mu$).
- D_Zmass: Masa invariante del par de leptones en el canal D ($\mu \mu \mu$).
- A_Dr_Z: Separación angular ΔR entre los leptones que forman el bosón Z en el canal A.
- B_Dr_Z: Separación angular ΔR entre los leptones que forman el Z en el canal B.
- C_Dr_Z: Separación angular ΔR entre los leptones que forman el Z en el canal C.
- D_Dr_Z: Separación angular ΔR entre los leptones que forman el Z en el canal D.
- MET_pt: Momento transversal faltante (p_T^{miss}), asociado a neutrinos no detectados.
- nElectron: Número de electrones reconstruidos en el evento.
- nMuon: Número de muones reconstruidos en el evento.

Estas variables potencialmente proporcionan separación entre procesos de señal y ruido lo cual es de relevancia al momento de entrenar la red autoencoder. En la figura 3.1 se muestran una gráfica de comparación entre proceso de ruido (Modelo Estándar) y señal (Wprime) para la variable MET_pt la cual esta relacionada a la energía faltante por la presencia del neutrino (proveniente del boson W) la cual se espera que el caso del SM sea una distribución con valores menores que la de la señal (Wprime), el resto de variables se muestran en el Apéndice A.

Antes de entrenar el AutoEncoder se analizaron los histogramas comparativos normalizados por densidad *señal* vs. *ruido* y la matriz de correlación como se muestra en la Figura 3.2.

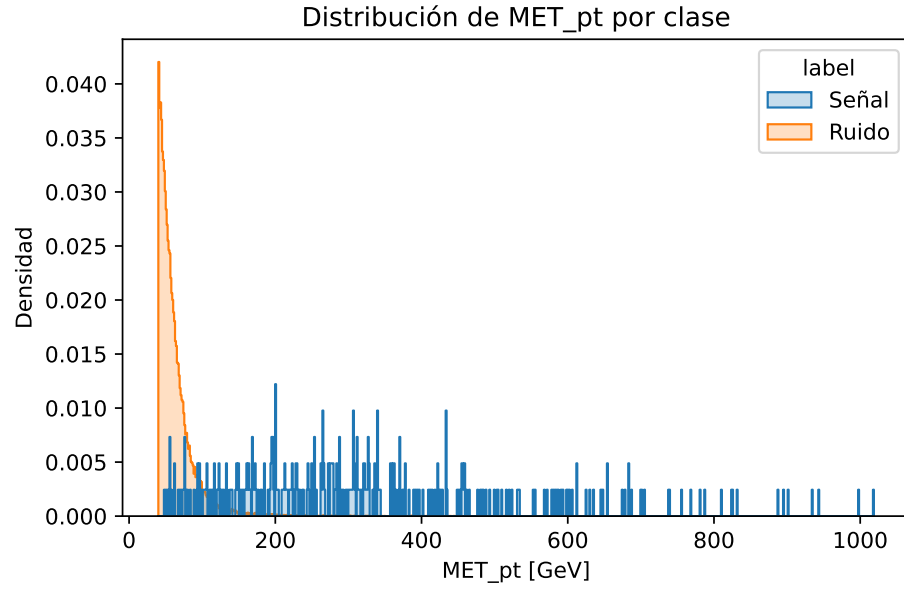


Figura 3.1: Distribución de MET_pt para eventos de ruido (SM) y señal.

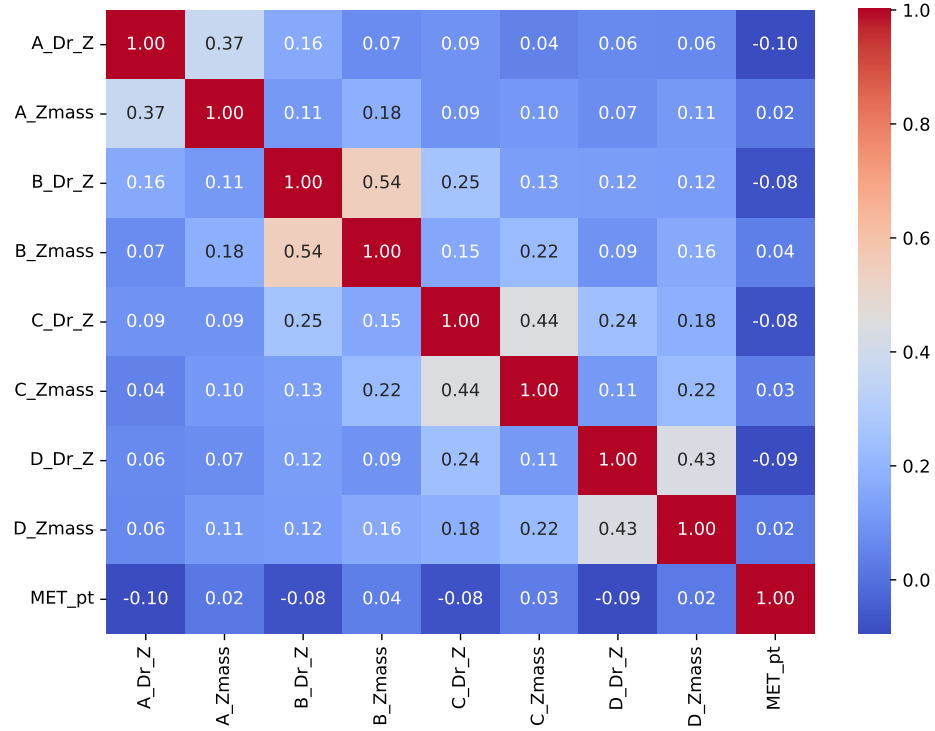


Figura 3.2: Matriz de correlación entre variables en la muestra de fondo.

Después de analizar la matriz de correlación y los histogramas comparativos, las cinco variables finalmente elegidas como **features** fueron:

- A_Dr_Z, B_Dr_Z, C_Dr_Z, D_Dr_Z (separaciones angulares en los canales A–D)
- MET_pt (momento transversal faltante)

3.3.2. Mezclado de eventos aleatorio

Usualmente, las muestras de simulación se encuentran separadas por tipos de procesos, como se describió en la tabla 3.1. Esto obedece al tipo de análisis que se realiza comúnmente en la física de altas energías; sin embargo, al alimentar una red neuronal se espera que la muestra usada represente tanto los procesos de ruido como los de señal.

Para tal efecto se realiza un proceso de mezclado (mixing) mediante el cual se genera un dataset combinado con todas las muestras y, de manera aleatoria, se reordenan los eventos; lo anterior busca reflejar la naturaleza aleatoria de los procesos físicos y recrear lo observado en los datos experimentales. Esto se logra gracias a un módulo de mezclado que se desarrolló íntegramente usando librerías de Python para el procesamiento de NanoAODs.

3.3.3. Conversión a formato hdf5

Con las muestras ya mezcladas de manera aleatoria, el último paso, antes de entrar al entrenamiento de la red neuronal, es convertir el formato de los datos. El formato original es ROOT, en el cual los datos se encuentran almacenados en una estructura tipo árbol (tree) con ramas (branches) y hojas (leaves), apropiada para los análisis del CMS y para los experimentos ligados al Gran Colisionador de Hadrones del CERN. Para efectos de entrenar la red neuronal, se convierte el formato ROOT a HDF5, que es un formato más general y compatible con las librerías de Python empleadas en el entrenamiento de redes neuronales.

Consecutivamente, la muestra total de datos se separa en tres particiones mutuamente exclusivas:

1. Dataset de entrenamiento 80 %
2. Dataset de validación 20 %
3. Prueba

Aunque un autoencoder puede entrenarse de manera no supervisada —es decir, sin requerir información sobre qué proceso originó cada evento—, en nuestro caso, al trabajar con muestras simuladas para el entrenamiento, sí contamos con esa información y podemos utilizarla para determinar si los datos corresponden a procesos de ruido o de señal.

Esto se hace mediante la variable `Dataset_ID`. Esta información es relevante, ya que nos permitirá evaluar qué tan bien se desempeña el modelo al clasificar anomalías en función del error de reconstrucción y del tipo de proceso. Además, podremos medir el desempeño del modelo mediante otros indicadores estadísticos, como las curvas ROC.

3.4. Entrenamiento de modelo autoencoder

El modelo implementado corresponde a un *autoencoder denso profundo* (*Deep Feed-Forward Auto-encoder*), cuyo objetivo es aprender la distribución multivariante de las variables seleccionadas en el paso de filtrado: MET_pt, A_Dr_Z, B_Dr_Z, C_Dr_Z, D_Dr_Z.

El bloque central del autoencoder se encuentra en el apéndice B.1.

Obsérvese la expansión inicial ($5 \rightarrow 64$) antes de comprimir: tal “sobrecompletitud” permite a la red proyectar los datos a un espacio de mayor expresividad donde la función SELU puede capturar

correlaciones no lineales antes de forzarlas al cuello de botella *"bottleneck"* de 16 neuronas. De acuerdo con el marco teórico (§2.3.2), esta topología en forma de *reloj de arena* maximiza la retención de información relevante y, simultáneamente, impide la copia trivial de la entrada.

Bloque	Capa	Dimensión	Activo / Norma
Entrada	—	5	—
Encoder	Linear	64	SELU + BN
	Linear	32	SELU + BN
Código latente	Linear	16	Lineal
Decoder	Linear	32	SELU + BN
	Linear	64	SELU + BN
Salida	Linear	5	Lineal

Tabla 3.2: Capas y activaciones del AutoEncoder

Cuando el modelo se entrena sólo con fondo (procesos habituales), se espera un error de reconstrucción bajo; un evento con nueva física (como W') mostraría un patrón cinemático que el AE no puede reproducir, dando un error alto.

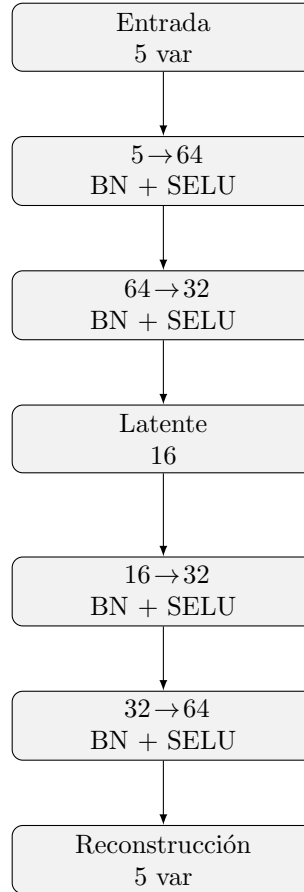


Figura 3.3: Diagrama del AutoEncoder empleado. El “cuello de botella” de 16 neuronas obliga a la red a retener en z sólo la información crucial de cada evento.

Umbral de reconstrucción.

Finalmente, se adopta un valor de 1300 para el error de reconstrucción como umbral único para discriminar eventos anómalos. Este valor se determina empíricamente observando la cola alta del histograma de errores y será utilizado en los análisis posteriores.

Capítulo 4

Resultados y discusión

4.1. Introducción

En este capítulo se presentan los resultados obtenidos tras entrenar y validar el autoencoder descrito anteriormente, empleando las cinco variables cinemáticas del Capítulo 3. Se analiza la evolución de la función de pérdida, la capacidad de discriminación medida mediante la curva ROC, la distribución del error de reconstrucción y la estructura del espacio latente proyectada sobre dos dimensiones.

4.2. Gráficas de entrenamiento

Una vez completado el ajuste del AutoEncoder se obtuvo un conjunto de gráficas que resumen de forma visual tanto la evolución del entrenamiento como la capacidad de discriminación del modelo. En la Fig. 4.1 se observa una disminución abrupta durante las primeras siete épocas hasta alcanzar valores de ~ 3 en el conjunto de entrenamiento y ~ 1 en validación, donde la curva se estabiliza. La separación moderada y estable entre ambas curvas indica que el modelo generaliza adecuadamente sin incurrir en sobreajuste.

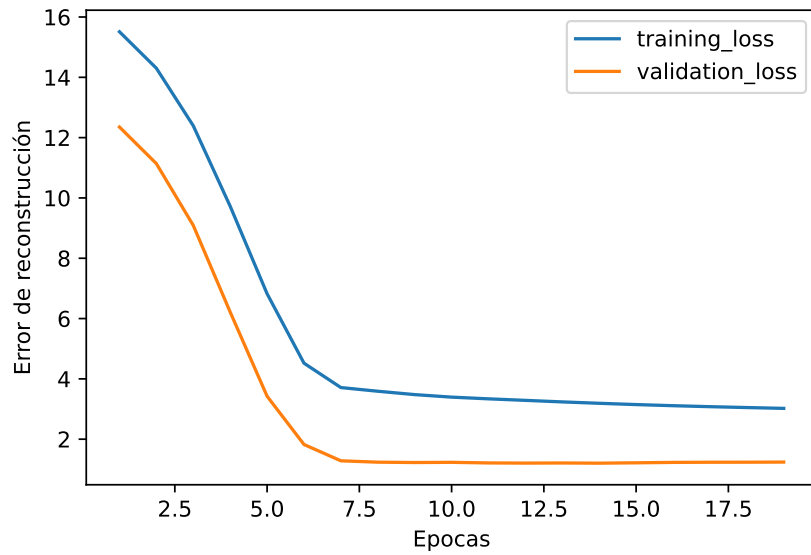


Figura 4.1: Curva de pérdida *train* / *validation*. La convergencia y la pequeña separación entre ambas curvas evidencian la ausencia de sobreajuste significativo durante las últimas épocas.

El desempeño global del detector de anomalías se resume en la Fig.4.2. La curva ROC asciende casi verticalmente, alcanzando valores de TVP superiores al 90 % cuando la tasa de falsos positivos es aún inferior al 1 %; el área bajo la curva resultante es $AUC = 0,99$, lo que confirma una capacidad de separación buena entre señal y fondo.

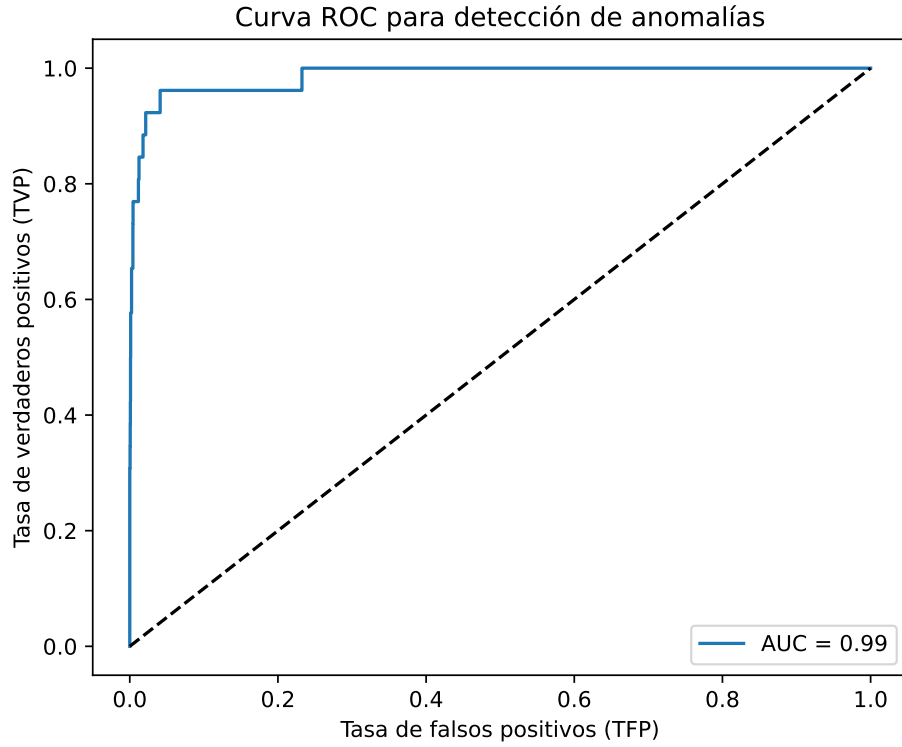


Figura 4.2: Curva ROC para la detección de anomalías mediante el error de reconstrucción. El área bajo la curva $AUC = 0,99$ demuestra alta capacidad de separación entre señal y fondo.

La interpretación cualitativa del espacio interno que construye el autoencoder se completa proyectando el código latente de dieciséis dimensiones a dos. En la Fig. 4.3 se muestran las dos primeras componentes principales: el fondo forma un haz curvado y denso cercano al origen, mientras que la señal se despliega a lo largo de la dirección de mayor varianza, separándose progresivamente del núcleo central. La proyección no lineal mediante t -SNE (Fig. 4.4) refuerza esta lectura, al situar los eventos de señal en una rama periférica del conjunto principal ocupado por el ruido. Ambas proyecciones evidencian que el modelo ha aprendido representaciones internas en las que la señal ocupa regiones escasamente pobladas por el fondo, coherentes con la definición de anomalía utilizada durante la validación cuantitativa.

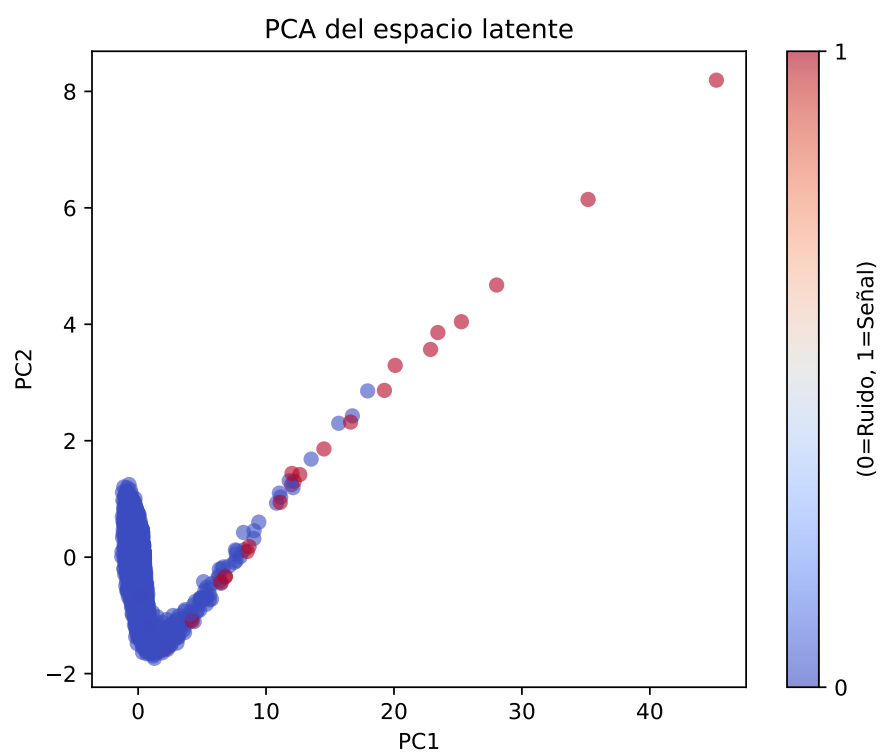


Figura 4.3: PCA del espacio latente. El fondo (azul) se agrupa en torno al origen, mientras que la señal (rojo) se dispersa hacia valores positivos de PC1 y PC2, lo que revela la presencia de componentes lineales que distinguen a los eventos anómalos.

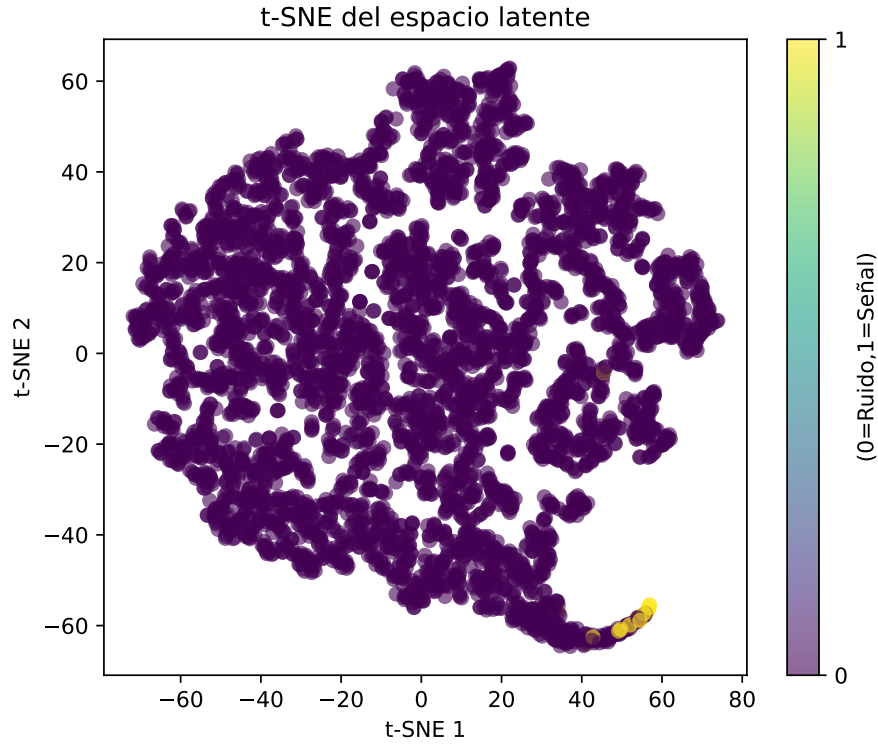


Figura 4.4: t-SNE del espacio latente. El fondo (morado) ocupa la mayor parte del espacio latente; los eventos de señal (amarillo) forman un racimo compacto en una región poco poblada, confirmando la capacidad del AE para aislar configuraciones atípicas.

Estos resultados gráficos proporcionan una validación visual de la metodología: confirman la convergencia del entrenamiento, la discriminación estadística entre clases y la pertinencia de la representación latente aprendida.

Las siguientes gráficas comparan la respuesta del modelo sobre eventos etiquetados como fondo frente a la muestra de señal sintetizada. Su correcta lectura complementa la información estadística previa y refuerza la intuición física del análisis.

Para comprender la procedencia de esta potencia discriminante se examina la distribución del error. La Fig.4.5 muestra el histograma en escala logarítmica; los eventos de fondo se concentran por debajo de 10^3 , mientras que la señal exhibe una cola larga hasta $\sim 4 \times 10^4$. A partir de esta observación se fijó un umbral $\tau = 1300$, valor que corresponde al percentil 95 de la distribución de fondo y que se utiliza para etiquetar un evento como anómalo. La Fig.4.6 representa el error frente al índice de evento con dicho umbral indicado mediante una línea discontinua; sobresalen veinticuatro instancias por encima del corte, quince de las cuales pertenecen a la señal simulada, lo que se traduce en una eficiencia cercana al 75 %. Cuando se magnifica la región de errores bajos (Fig. 4.7) se confirma que, por debajo de τ , la señal es prácticamente inexistente, lo que valida la elección del umbral.

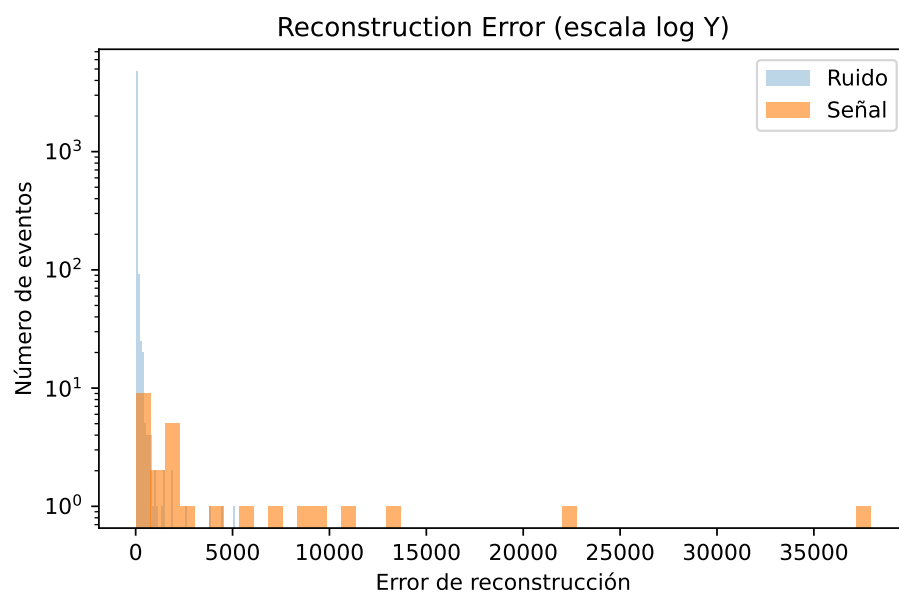


Figura 4.5: Histograma del error de reconstrucción. La señal W' (naranja) se desplaza a valores de error mayores que el fondo (azul), indicando que el AE reconstruye peor los eventos anómalos.

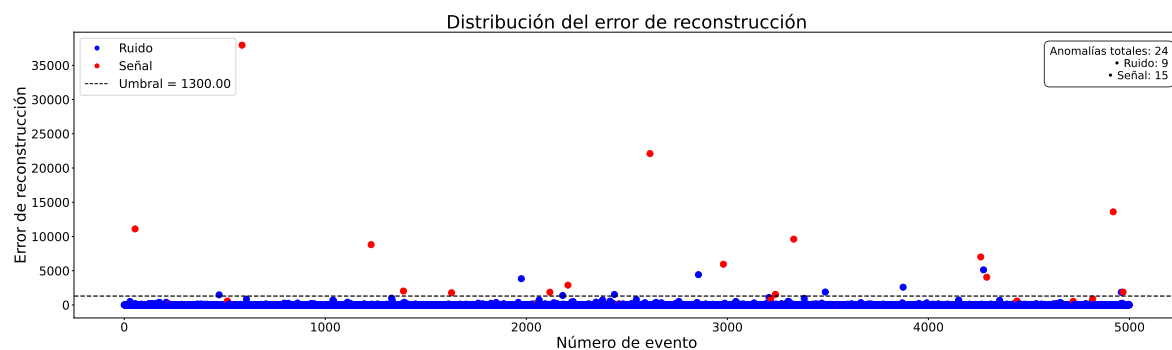


Figura 4.6: Dispersión del error de reconstrucción por evento. Arriba: escala completa donde la señal (rojo) sobresale de acuerdo al umbral establecido

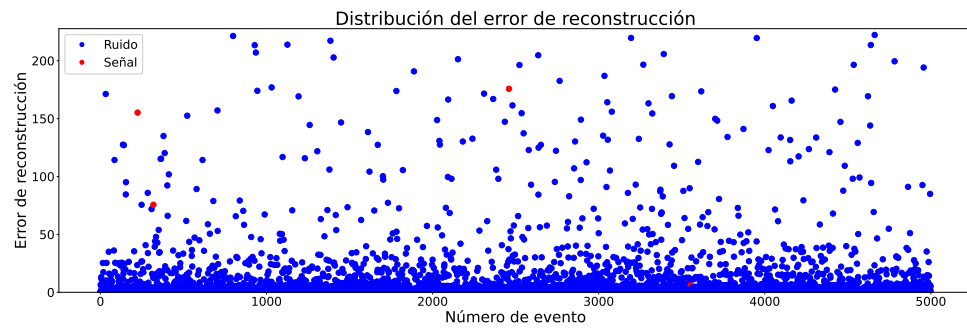


Figura 4.7: Ampliación de la banda donde se puede apreciar la predominancia del ruido (azul).

Capítulo 5

Conclusiones

Se entrenó un modelo de red neuronal autoencoder para detectar anomalías en eventos caracterizados por tres o más leptones en el contexto del experimento CMS del CERN. Las muestras usadas para el entrenamiento son procesos del Modelo Estándar que se caracterizan por la producción de tres o más leptones (ZZ , WZ) y un proceso hipotético de señal ($W' \rightarrow WZ \rightarrow 3l$), el cual sirve como referencia para caracterizar las posibles anomalías en los datos.

El método consistió en un procesamiento de datos para preparar las muestras que alimentan la red neuronal autoencoder, la validación mediante un conjunto de datos independiente y la extracción de resultados. Se definió un umbral en el error de reconstrucción del autoencoder de $\tau = 1300$; con él se separan los eventos 'normales' de las anomalías. De los resultados se observa que, en la mayoría de los casos, el modelo identifica los eventos de señal como anomalías, lo que aporta confianza sobre su posible uso con datos experimentales. En una continuación de este estudio, el modelo puede aplicarse a datos experimentales (usando un disparador de tres leptones) para evaluar la selección de eventos y caracterizar los datos seleccionados en la búsqueda de señales de nueva física.

Como proyección inmediata, se planea ejecutar el autoencoder de forma no supervisada sobre muestras del *Open Data* del CMS, de manera que los propios datos determinen la región de ruido y señal sin recurrir a simulaciones. Los eventos marcados como anómalos se inspeccionarán visualmente con *iSpy WebGL*, un software oficial del CERN para la visualización interactiva de colisiones. Este paso permitirá corroborar si las anomalías corresponden a topologías físicas coherentes o a artefactos de reconstrucción y, en el mejor de los casos, abrirá la posibilidad de realizar búsquedas dirigidas de nueva física con un enfoque plenamente basado en datos experimentales.

Referencias

- [1] B. Moser, *The Beauty and the Boost: A Higgs Boson Tale, Measurements of Higgs Boson Production at High Energy in Decays to Bottom Quarks and Their Interpretations with the ATLAS Experiment at the LHC*. Springer Nature, 2023.
- [2] K. Albertsson et al., “Machine Learning in High Energy Physics Community White Paper,” en *Journal of Physics: Conference Series*, vol. 1085, IOP Publishing, sep. de 2018, pág. 022008. DOI: 10.1088/1742-6596/1085/2/022008.
- [3] M. A. Pimentel, D. A. Clifton, L. Clifton y L. Tarassenko, “A Review of Novelty Detection,” *Signal Processing*, vol. 99, págs. 215-249, 2014. DOI: 10.1016/j.sigpro.2013.12.026.
- [4] V. Belis, P. Odagiu y T. K. Aarrestad, “Machine Learning for Anomaly Detection in Particle Physics,” *Reviews in Physics*, pág. 100091, 2024. DOI: 10.1016/j.revip.2023.100091.
- [5] T. C. Collaboration, “The CMS Experiment at the CERN LHC,” *Journal of Instrumentation*, vol. 3, n.º 08, S08004, 2008. DOI: 10.1088/1748-0221/3/08/S08004.
- [6] “Autoencoder-Based Anomaly Detection System for Online Data Quality Monitoring of the CMS Electromagnetic Calorimeter,” *Computing and Software for Big Science*, vol. 8, n.º 1, pág. 11, 2024.
- [7] N. Zipper et al., “Testing a Neural Network for Anomaly Detection in the CMS Global Trigger Test Crate During Run 3,” *Journal of Instrumentation*, vol. 19, n.º 03, pág. C03029, 2024. DOI: 10.1088/1748-0221/19/03/C03029.
- [8] C. Luo y A. Shrivastava, “Arrays of (Locality-Sensitive) Count Estimators (ACE): High-Speed Anomaly Detection via Cache Lookups,” *arXiv preprint*, vol. abs/1706.06664, 2017. arXiv: 1706.06664 [cs.LG].
- [9] D. Guest, K. Cranmer y D. Whiteson, “Deep Learning and its Application to LHC Physics,” *Annual Review of Nuclear and Particle Science*, vol. 68, n.º 1, págs. 161-181, 2018. DOI: 10.1146/annurev-nucl-101917-021019.
- [10] A. Collaboration, “The ATLAS Experiment at the CERN Large Hadron Collider,” *Journal of Instrumentation*, vol. 3, S08003, 2008. DOI: 10.1088/1748-0221/3/08/S08003.
- [11] G. Polesello, “Introduction to LHC Physics,” en *Journal of Physics: Conference Series*, vol. 53, IOP Publishing, nov. de 2006, pág. 107. DOI: 10.1088/1742-6596/53/1/107.
- [12] S. F. Novaes, “Standard Model: An Introduction,” en *10th Jorge Andre Swieca Summer School: Particle and Fields*, 2000, págs. 5-102.
- [13] Wikimedia Commons contributors, *Standard Model of Elementary Particles*, https://upload.wikimedia.org/wikipedia/commons/thumb/0/00/Standard_Model_of_Elementary_Particles.svg/2140px-Standard_Model_of_Elementary_Particles.svg.png, Licencia CC BY 3.0, 2024.
- [14] C. Collaboration et al., “Identification of Heavy-Flavour Jets with the CMS Detector in pp Collisions at 13 TeV,” *Journal of Instrumentation*, vol. 13, n.º 5, P05011, 2018. DOI: 10.1088/1748-0221/13/05/P05011.

- [15] C. Collaboration, “Performance of the CMS missing transverse momentum reconstruction in pp data at $\sqrt{s} = 8\text{ TeV}$,” *JINST*, vol. 10, n.º 02, P02006, 2015. DOI: 10.1088/1748-0221/10/02/P02006. arXiv: 1411.0511 [hep-ex].
- [16] C. collaboration, “Search for new resonances decaying via WZ to leptons in proton-proton collisions at $\sqrt{s} = 8\text{ TeV}$,” *Physics Letters B*, 2015. DOI: 10.1016/j.physletb.2014.11.026.
- [17] Y. Bastanlar y M. Ozuysal, “Introduction to Machine Learning,” en *miRNomics: MicroRNA Biology and Computational Analysis*, M. Yousef y J. Allmer, eds., Humana Press, 2014, págs. 105-128. DOI: 10.1007/978-1-62703-748-8_7.
- [18] K. Gurney, *An Introduction to Neural Networks*. CRC Press, 2018, ISBN: 9781138468590.
- [19] A. D. Dongare et al., “Introduction to Artificial Neural Network,” *International Journal of Engineering and Innovative Technology (IJEIT)*, vol. 2, n.º 1, págs. 189-194, 2012.
- [20] R. Y. Choi et al., “Introduction to Machine Learning, Neural Networks, and Deep Learning,” *Translational Vision Science & Technology*, vol. 9, n.º 2, págs. 14-14, 2020. DOI: 10.1167/tvst.9.2.14.
- [21] S. Grossberg, “Recurrent Neural Networks,” *Scholarpedia*, vol. 8, n.º 2, pág. 1888, 2013. DOI: 10.4249/scholarpedia.1888.
- [22] U. Michelucci, “An Introduction to Autoencoders,” *arXiv preprint*, 2022. eprint: arXiv:2201.03898.
- [23] G. Klambauer, T. Unterthiner, A. Mayr y S. Hochreiter, “Self-Normalizing Neural Networks,” en *Advances in Neural Information Processing Systems 30 (NeurIPS 2017)*, Curran, 2017, págs. 971-980. DOI: 10.48550/arXiv.1706.02515. arXiv: 1706.02515 [cs.LG].
- [24] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA: MIT Press, 2012, ISBN: 9780262018029.
- [25] T. Heimel, G. Kasieczka, T. Plehn y J. M. Thompson, “QCD or What?” *SciPost Phys.*, vol. 6, n.º 3, pág. 030, 2019. DOI: 10.21468/SciPostPhys.6.3.030.
- [26] M. Farina, Y. Nakai y D. Shih, “Searching for New Physics with Deep Autoencoders,” *Phys. Rev. D*, vol. 101, n.º 7, pág. 075 021, 2020. DOI: 10.1103/PhysRevD.101.075021.
- [27] L. van der Maaten y G. E. Hinton, “Visualizing Data Using t-SNE,” *J. Mach. Learn. Res.*, vol. 9, págs. 2579-2605, 2008.
- [28] R. Brun y F. Rademakers, “ROOT: An Object Oriented Data Analysis Framework,” *Nucl. Instrum. Meth. A*, vol. 389, págs. 81-86, 1997. DOI: 10.1016/S0168-9002(97)00048-X.
- [29] G. Petrucciani, A. Rizzi, C. Vuosalo y C. Collaboration, “Mini-AOD: A New Analysis Data Format for CMS,” *J. Phys. Conf. Ser.*, vol. 664, pág. 072 052, 2015. DOI: 10.1088/1742-6596/664/7/072052.
- [30] A. Rizzi, G. Petrucciani, M. Peruzzi y C. Collaboration, “A Further Reduction in CMS Event Data for Analysis: the NanoAOD Format,” *EPJ Web Conf.*, vol. 214, pág. 06 021, 2019. DOI: 10.1051/epjconf/201921406021.
- [31] J. H. Collins, K. Howe y B. Nachman, “Extending the Search for New Resonances with Machine Learning,” *Phys. Rev. D*, vol. 99, n.º 1, pág. 014 038, 2019. DOI: 10.1103/PhysRevD.99.014038.
- [32] E. Gross y O. Vitells, “Trial Factors for the Look Elsewhere Effect in High Energy Physics,” *Eur. Phys. J. C*, vol. 70, págs. 525-530, 2010. DOI: 10.1140/epjc/s10052-010-1470-8.

Apéndice A

Comparación ruido y señal

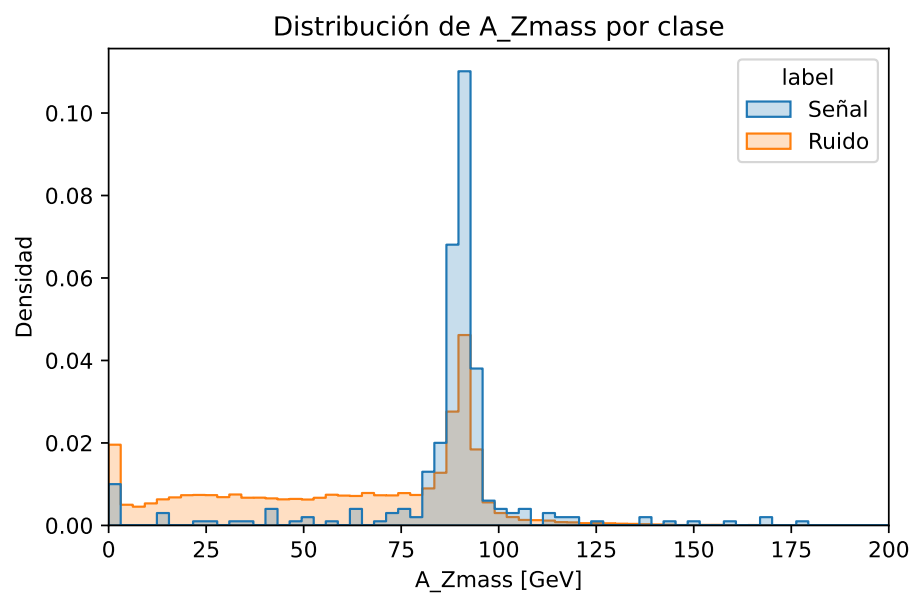


Figura A.1: Distribución de A_Zmass por clase.

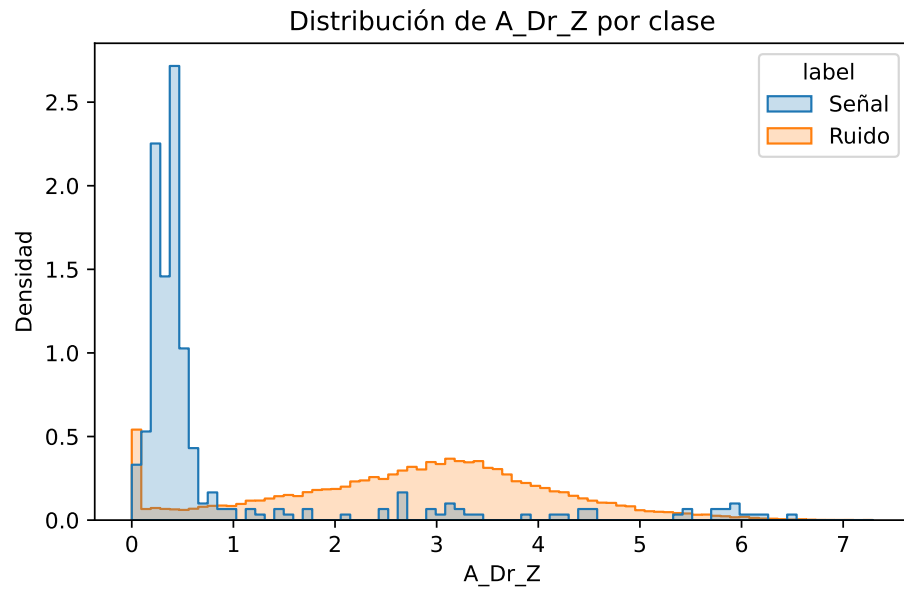


Figura A.2: Distribución de A_Dr_Z por clase.

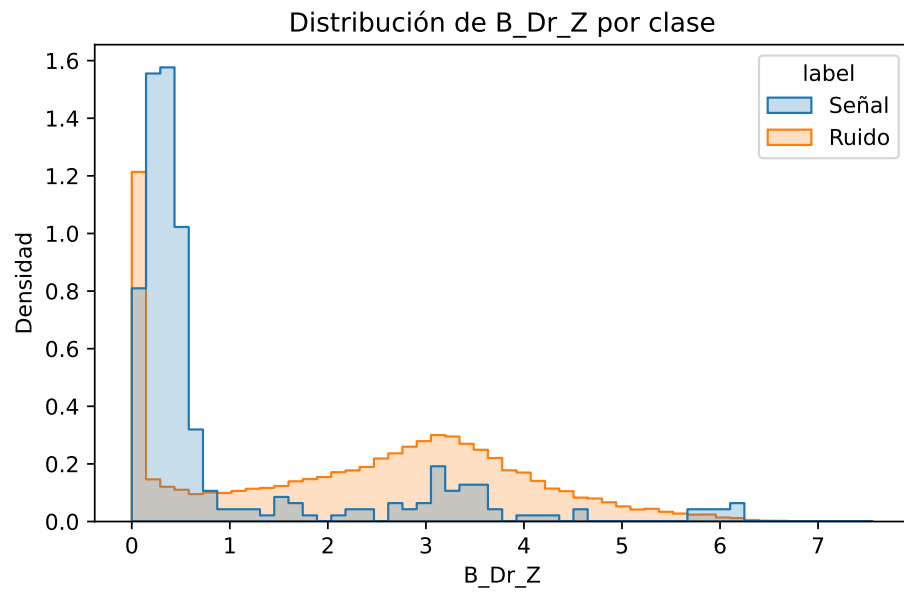


Figura A.3: Distribución de B_Dr_Z por clase.

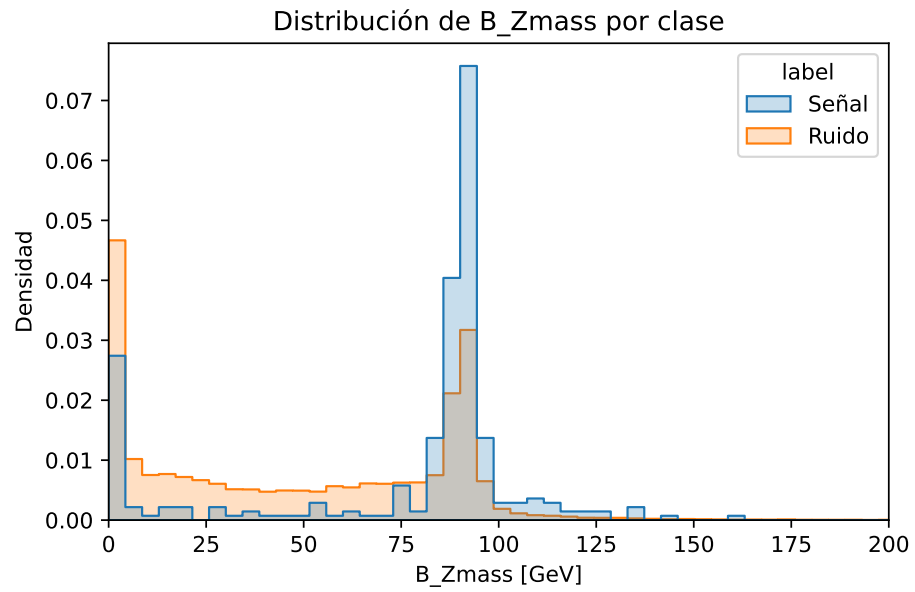


Figura A.4: Distribución de B_Zmass por clase.

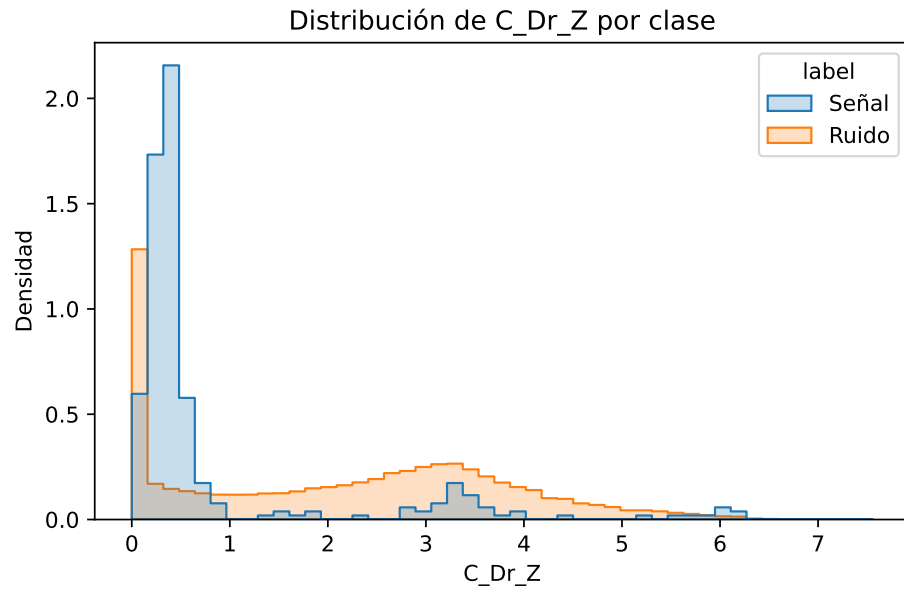


Figura A.5: Distribución de C_Dr_Z por clase.

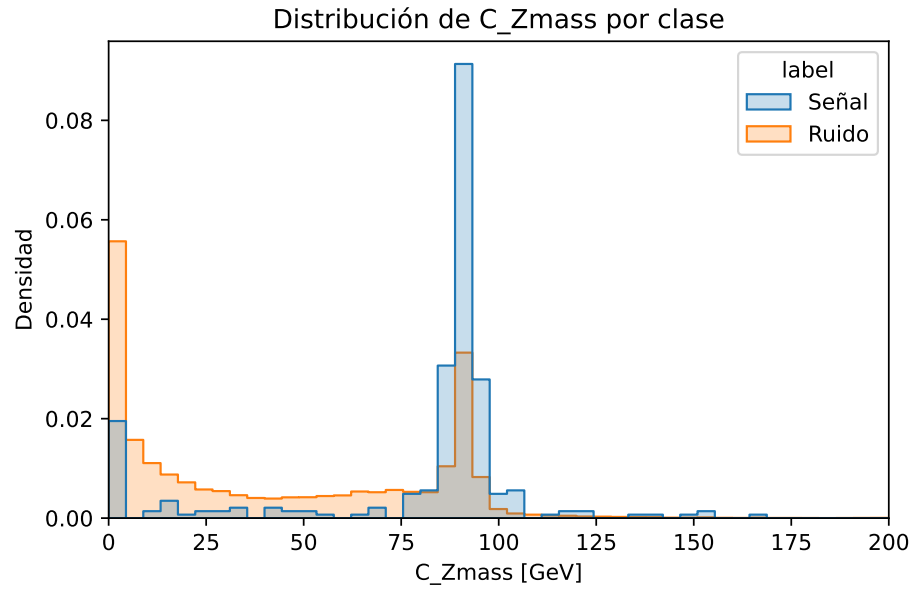


Figura A.6: Distribución de C_{Zmass} por clase.

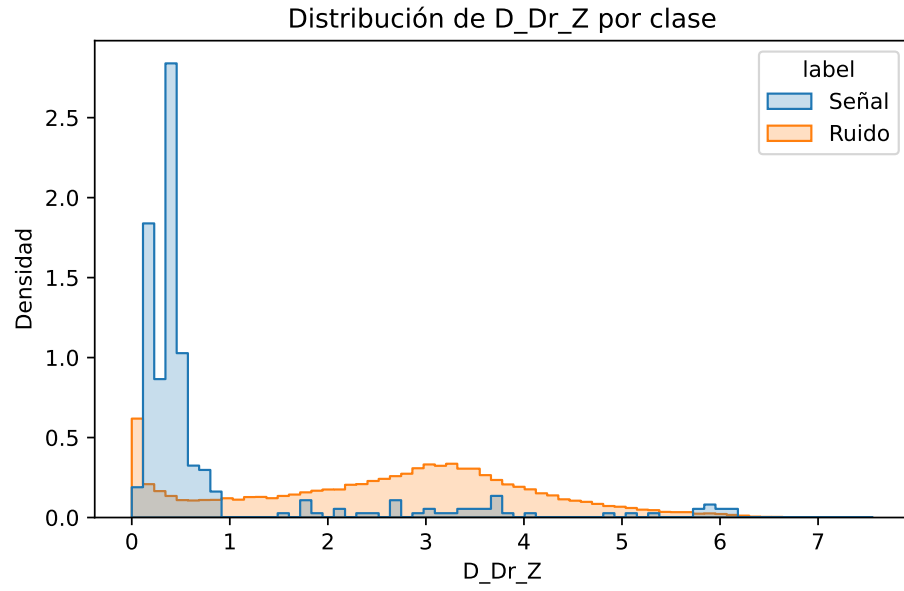


Figura A.7: Distribución de D_{Dr_Z} por clase.

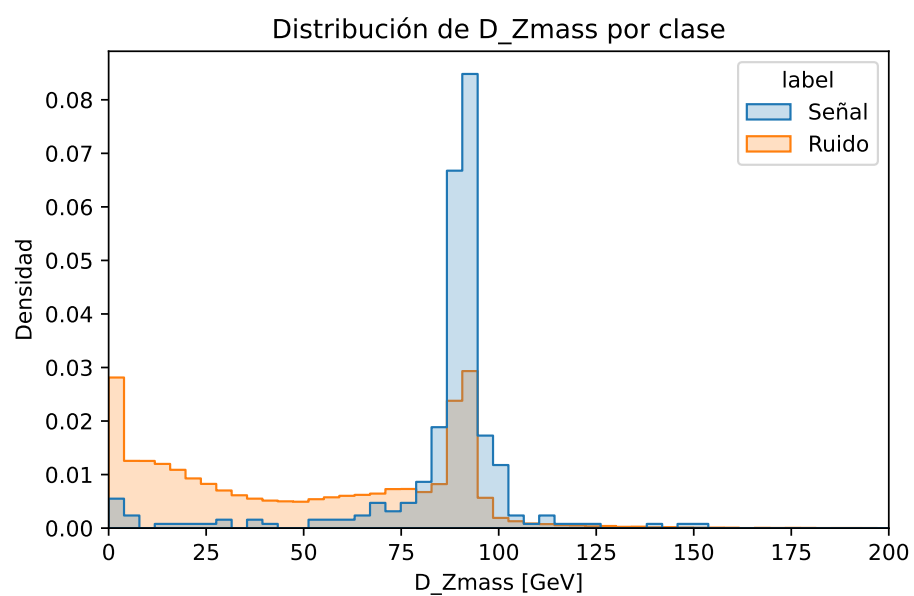


Figura A.8: Distribución de D_Zmass por clase.

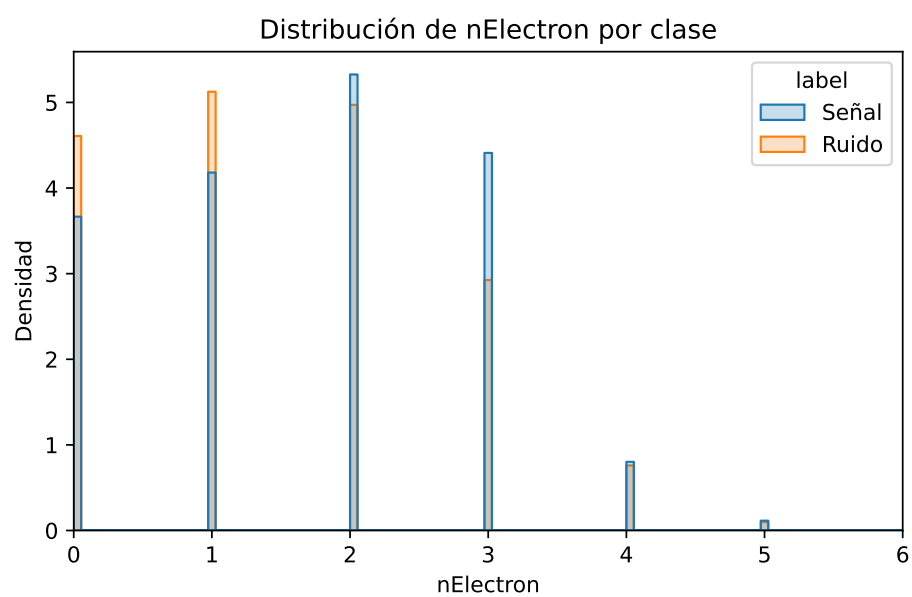


Figura A.9: Distribución de nElectron por clase.



Figura A.10: Distribución de **nMuon** por clase.

Apéndice B

Secciones de código

B.1. Bloque central del AutoEncoder

El *encoder* expande primero la entrada de 27 variables a 64 neuronas, aplica normalización por lote y función SELU; después la comprime de forma gradual hasta un código latente de 16 unidades. El *decoder* invierte simétricamente la transformación para reconstruir la entrada. Se incluye la propagación hacia adelante, pasos de entrenamiento, validación, predicción y configuración del optimizador. Este fragmento muestra cómo se guarda la configuración, se construyen las secuencias de capas, y cómo se registran y recolectan las pérdidas durante el entrenamiento y la validación.

```
class Autoencoder(L.LightningModule):
    def __init__(self, in_dim):
        super().__init__()
        self.save_hyperparameters()

        self.encoder = nn.Sequential(
            nn.Linear(in_dim, 64),
            nn.BatchNorm1d(64),
            nn.SELU(),
            nn.Linear(64, 32),
            nn.BatchNorm1d(32),
            nn.SELU(),
            nn.Linear(32, 16)
        )

        self.decoder = nn.Sequential(
            nn.Linear(16, 32),
            nn.BatchNorm1d(32),
            nn.SELU(),
            nn.Linear(32, 64),
            nn.BatchNorm1d(64),
            nn.SELU(),
            nn.Linear(64, in_dim)
        )

        self.training_losses = []
        self.validation_losses = []

    def forward(self, x):
        z = self.encoder(x)
        return self.decoder(z)
```

```

def training_step(self, batch, batch_idx):
    x      = batch
    x_hat  = self(x)
    loss   = F.smooth_l1_loss(x_hat, x)
    self.log('train_loss', loss, prog_bar=True)
    return loss

def on_train_epoch_end(self):
    self.training_losses.append(
        self.trainer.callback_metrics['train_loss'].item()
    )

def validation_step(self, batch, batch_idx):
    input  = batch
    output = self.forward(input)
    loss   = F.smooth_l1_loss(output, input)
    self.log('val_loss', loss, prog_bar=True)
    return loss

def on_validation_epoch_end(self):
    self.validation_losses.append(
        self.trainer.callback_metrics['val_loss'].item()
    )

def predict_step(self, batch, batch_idx, dataloader_idx=None):
    input  = batch
    output = self.forward(input)
    return output

def configure_optimizers(self):
    return torch.optim.AdamW(
        self.parameters(),
        lr=1e-4,
        weight_decay=1e-5
    )

```

B.2. Configuración del Trainer y callbacks

Fragmento que muestra la configuración de entrenamiento: *early stopping*, *model checkpoint*, monitor de tasa de aprendizaje. El trainer limita el número máximo de épocas a 1000, se detiene si no hay mejora significativa en 5 épocas y permite acelerador automático (GPU si está disponible).

```

es      = EarlyStopping(monitor="val_loss",
                        mode="min", min_delta=1e-5, patience=5)
cp      = ModelCheckpoint(save_top_k=1,
                        monitor="val_loss", mode="min")
lr_monitor = LearningRateMonitor(logging_interval="step")

trainer = L.Trainer(
    callbacks=[SimpleProgressBar(), es, cp, lr_monitor],
    max_epochs=1000,
    enable_checkpointing=True,
    accelerator="auto"
)

```

```
trainer.fit(model, train_dataloaders=ae_tdl,  
            val_dataloaders=ae_vdl)
```