

Primera vista de correlaciones Bose-Einstein para piones con datos experimentales de CLAS12, Jefferson Lab.

Nicolás Araya P.

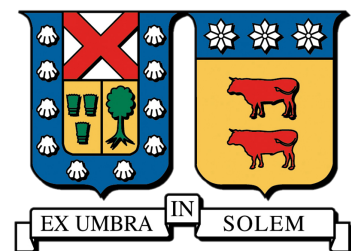
Tesina para optar al grado

Licenciatura en ciencias mencion física

marzo 2025

Supervisor: Prof. Hayk Hakobyan

Departamento de física



Abstract

This undergraduate research dissertation analyzes data from the CLAS12 spectrometer at Jefferson Lab's RG-E experiment using a 12 GeV electron beam to identify positive pions and study Bose-Einstein correlations. Preliminary results are compared to the EG2 experiment (6 GeV electron beam) to explore the DIS of the electron phase space. Data analysis is performed in ROOT with C++.

En esta tesina de investigación de grado, se analiza datos del espectrómetro CLAS12 en el experimento RG-E del Jefferson Lab, utilizando un haz de electrones de 12 GeV, con el objetivo de identificar piones positivos y estudiar las correlaciones de Bose-Einstein. Los resultados preliminares se comparan con el experimento EG2 (haz de 6 GeV) para explorar el DIS del espacio de fases de los electrones. El análisis de datos se realiza en ROOT utilizando C++.

Agradecimientos

Agradezco a mis padres quienes me han apoyado incondicionalmente durante toda mi vida académica. Agradezco también a mis compañeros y pares que me han inspirado y guiado. Agradezco a Sofía por todo el amor que me ha entregado.

Abreviaciones

Abreviación	Descripción
BEC	Bose Einstein Correlations
QCD	Quantum Chromodynamics
QED	Quantum Electrodynamics
DIS	Deep Inelastic Scattering
SIDIS	Semi Inclusive Deep Inelastic Scattering
CEBAF	Continuous Electron Beam Accelerator Facility
CLAS	CEBAF Large Acceptance Spectrometer

List of Figures

1.1	Modelo estándar de física de partículas (Cush, 2025). Public domain image from Wikimedia Commons.	3
1.2	SIDIS para un leptón cualquiera impactando en un hadrón cualquiera	8
1.3	La trayectoria de dos piones positivos indistinguibles	9
1.4	Diseño sistema double target usado en CLAS12	14
1.5	Sistema de Double Target	14
2.1	Un histograma respecto a la variable v_z para observar los target.	21
2.2	Una distribución Q2 sobre x_b del archivo ntuples_dc_020082.root	22
2.3	Los cortes DIS aplicados de la figura 2.2	22
3.1	Distribuciones <i>signal</i> en rojo, <i>background</i> en azul para Deuterio (D2) con 50 bins	28
3.2	Distribuciones <i>signal</i> en rojo, <i>background</i> en azul para Carbono (C) con 50 bins	28
3.3	Distribuciones <i>signal</i> en rojo, <i>background</i> en azul para Aluminio (Al) con 50 bins	29
3.4	Distribuciones <i>signal</i> en rojo, <i>background</i> en azul para Cobre (Cu) con 50 bins	29
3.5	Distribuciones <i>signal</i> en rojo, <i>background</i> en azul para Estaño (Sn) con 50 bins	30
3.6	Distribuciones <i>signal</i> en rojo, <i>background</i> en azul para Plomo (Pb) con 50 bins	30
3.7	Función de correlación para el Deuterio (D2) con 50 bins	32
3.8	Función de correlación para el Carbono (C) con 60 bins	32
3.9	Función de correlación para el Aluminio (Al) con 40 bins	33
3.10	Función de correlación para el Cobre (Cu) con 40 bins	33
3.11	Función de correlación para el Estaño (Sn) con 60 bins	34
3.12	Función de correlación para el Plomo(Pb) con 40 bins	34
3.13	Los valores de λ y r_G para el <i>target</i> más liviano en cada una de las colaboraciones incluida EG-2 (valor por Antonio Radic) y RG-E (de esta tesina). Una linea punteada está fija en 0.5 como referencia.	37
3.14	Los valores de r_G para los distintos <i>target</i>	37

List of Tables

2.1	En la tabla se presentan los archivos usados, con sus numeros de <i>run</i> y los <i>target</i> asociados.	17
3.1	Los valores de los parámetros con el ajuste de Goldhaber. Los valores con asterisco están limitados a 1.	35
3.2	Resultados de estadística χ^2 para diferentes <i>bins</i> en el intervalo de T [0, 1.6] <i>GeV</i>	36

Contents

Abstract	iii
Agradecimientos	iv
Abreviación	v
List of Figures	vi
List of Tables	vii
Contents	viii
1 Introducción	1
1.1 Conceptos Clave y Definiciones	4
1.1.1 Medio Nuclear y QCD	4
1.1.2 Semi Inclusive Deep Inelastic Scattering	7
1.1.3 Bose Einstein Correlations	8
1.1.4 Estudios Previos	11
1.2 Experimento RG-E y CLAS12	13
2 Método	15
2.1 Conceptos ROOT y c++	15
2.2 Datos Preprocesados	16
2.3 Procesamiento de datos	20
2.3.1 Cortes/cuts	20
2.3.2 Pares de piones	22
2.3.3 Signal & background	23
2.3.4 Calculo Correlación	25
3 Análisis de Resultados	27
4 Discusión y Conclusiones	38
4.1 Discusión	38
4.1.1 Explicaciones para las incertidumbres	39
4.2 Conclusion	40

Bibliografía	43
---------------------	-----------

Anexos	43
---------------	-----------

A Anexo A — Códigos C++	45
A.1 FilterNtupleDIS.C	45
A.2 MergeNtuples.C	47
A.3 FilterPairsMerged.C	48
A.4 expT.C	50
A.5 MEMspherical.C	52
A.6 hratioBEC3.C	55

1

Introducción

A principios de siglo XX, la física tendría una revolución en la forma que se entiende debido a teorías que se construyeron a lo largo de varias décadas como la de la relatividad y mecánica cuántica. Varias ideas se construyeron y se rechazaron, cada una con la evidencia empírica correspondiente. Como resultado, estos avances científicos se extendieron hacia otras ramas de la ciencia, como astrofísica, química y biología, aumentando nuestro conocimiento del universo. Es la labor de científicos, ingenieros e investigadores contemporáneos continuar esta trayectoria e ir más profundo de lo que se pensó posible. Es con esta idea que se introduce la física de partículas de altas energías, la rama que explora los límites de esta ciencia a velocidades muy superiores a la de la vida cotidiana y a tamaños más pequeños que un átomo.

La física de partículas investiga la materia nuclear subatómica y las fuerzas fundamentales del universo. En los años 70, se construyó una teoría que describe el comportamiento y dinámica de tres de las cuatro fuerzas fundamentales, y todas las partículas elementales llamada **Modelo Estándar**. En la figura 1.1) se ve el modelo estándar en su forma más actualizada, incluyendo el bosón de Higgs, descubierto el año 2012. La primera columna del lado izquierdo (fermiones) son conocidas como la primera generación y componen la mayoría de la materia normal del universo. En naranja están los bosones de gauge, que describen las interacciones. Este modelo describe las partículas elementales y las fuerzas

en 3 de las 4 interacciones fundamentales del universo¹ y para describir estas fuerzas fundamentales, se construyeron las siguientes teorías: **Fuerza Electromagnética** abreviada *QED, Quantum Electrodynamics*, **Fuerza Débil** *QFD Quantum Flavordynamics* y **Fuerza Fuerte** *QCD, Quantum Chromodynamics*. De la figura 1.1) los **quarks**, que constituyen a otras partículas, como protones, y las fuerzas están representadas por partículas mediadoras, llamadas **bosones de gauge**.

En particular, **QCD** es la teoría de las interacciones fuertes en el modelo estándar y explica como los neutrones y protones están ligados o confinados en el núcleo² y actualmente no existe una descripción teórica y analítica de algunos de sus comportamientos, solamente fenomenológica. Una de las propiedades de interés es la **hadronización**, un proceso que ocurre lejos del núcleo y no puede ser abordado con un análisis perturbativo, es decir usar aproximaciones para describir sistemas cuánticos a partir de otros más sencillos.

La hadronización consiste en la formación de hadrones, una partícula subatómica formada por quarks y permanecen unidos por la interacción fuerte. Este fenómeno está directamente relacionado al confinamiento del color, una propiedad se observada con suficiente evidencia [2] y proviene de que empíricamente jamás se ha observado un quark o gluón libre a bajas energías. Imagínese por un momento que un quark es “arrancado” del núcleo, como en un proceso de **scattering** de altas energías. La energía que necesita para ser arrancado aumenta linealmente como si fuese una cuerda de energía. Si se sigue tirando el quark, eventualmente esta cuerda se romperá y de cada extremo surgen dos nuevos hadrones debido a la creación espontánea de un quark y anti-quark. Este es el modelo **Lund String**, una descripción fenomenológica de la hadronización.

La hadronización es de gran interés en la investigación de la teoría *QCD*, y se puede observar en procesos *Quark Gluon Plasma* en colisiones pesadas y relativistas como las llevada a cabo en *ALICE* en CERN, o también en *color string decay* que puede ser explorado mediante *Deep Inelastic Scattering* o **DIS**, como el

¹El modelo estándar no incluye a la fuerza fundamental gravitatoria, pues no hay una descripción satisfactoria dentro del modelo, como si lo hay en relatividad general. Se ha propuesto una particular teórica llamada gravitón, pero no hay evidencia de esta partícula.

²99.9 % de la masa de la materia ordinaria está constituida de protones y neutrones [1]

Modelo estándar de física de partículas

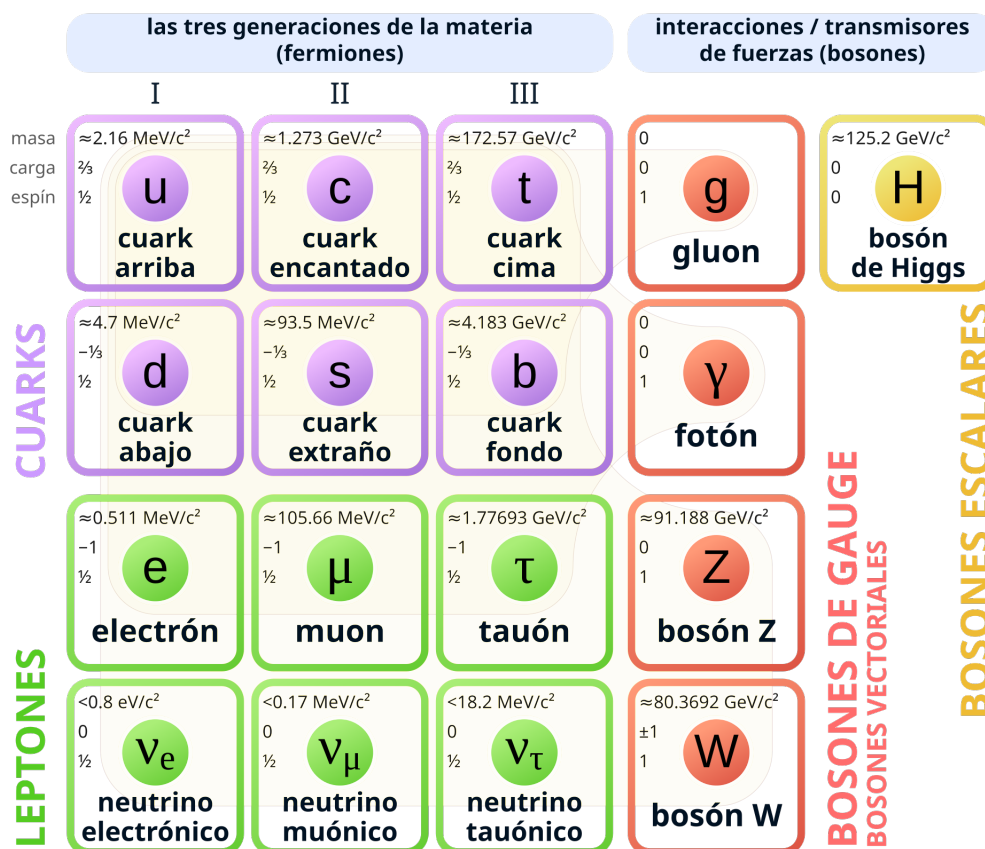


Figure 1.1: Modelo estándar de física de partículas (Cush, 2025). Public domain image from Wikimedia Commons.

que se implementa en el experimento **RG-E** de *Jefferson Lab*, en el cual la investigación presente obtiene sus datos. Entre marzo y mayo de 2024, en *Jefferson Lab*, Virginia, se realizaron una serie de experimentos en el acelerador de electrones **Continuous Electron Beam Accelerator Facility (CEBAF)**, como parte del experimento **RG-E**. En este experimento, se utilizó un haz de electrones de 12 GeV para estudiar procesos de **Deep Inelastic Scattering (DIS)** en núcleos objetivo. El uso de una energía tan alta permite explorar el comportamiento de los quarks y gluones en el régimen de alta transferencia de momento (Q^2), revelando aspectos fundamentales de la interacción fuerte. En particular, uno de los objetivos clave de este experimento es identificar piones positivos resultantes de la DIS y estudiar su producción analizando distribuciones hechas por pares de piones que siguen la estadística de Bose-Einstein.

El análisis de los piones positivos es de gran relevancia debido a que, al ser bosones, obedecen las estadísticas de *Bose-Einstein*, lo que posibilita el estudio de las **correlaciones Bose-Einstein** (*BEC*³). Estas correlaciones permiten obtener información sobre el tamaño y la duración de la fuente emisora de partículas, lo que brinda una ventana única hacia la comprensión del mecanismo de hadronización y las propiedades de la fase final de las colisiones. Además, el experimento **RG-E** con el espectrómetro CLAS12 ofrece una mejora significativa en términos de energía y resolución comparado con experimentos previos, como el experimento **EG2**⁴, que utilizó un haz de 6 GeV. Esta diferencia en la energía de los haces posibilita una comparación directa entre los resultados obtenidos a distintas escalas de energía, lo que podría permitir validar modelos fenomenológicos como el modelo de *Lund String* y obtener una mejor comprensión de los procesos de hadronización en diferentes condiciones experimentales.

Este análisis preliminar de las correlaciones Bose-Einstein (*BEC*) en **RG-E** proporcionará nueva evidencia sobre la dinámica de las interacciones hadrónicas y sentará las bases para futuras investigaciones en este campo. Además, permitirá una mejor estimación del espacio de fase accesible en la DIS, lo que contribuirá al avance en el estudio de la estructura interna de los nucleones y de las propiedades emergentes de las interacciones de quarks y gluones incluso a bajas energías.

1.1 Conceptos Clave y Definiciones

A continuación se definirán conceptos clave como marco teórico.

1.1.1 Medio Nuclear y QCD

El principal actor de la física moderna es el átomo, la unidad fundamental de la química y compuesto por un núcleo y uno o varios electrones que lo “orbitan”. Pero es necesario explicar y profundizar en esta descripción.

³No confundir con *Bose-Einstein Condensate*

⁴Gran parte de esta investigación fue gracias a la ayuda de Antonio Radic, a quien le debo agradecimientos especiales [3]

Un átomo tiene un tamaño aproximado de 0.5-3 ångstrom de radio con un ångstrom igual a 100 pm o 10^{-10} m. En cambio su núcleo es ridículamente pequeño⁵ con 1 fm o 10^{-15} m. Es en este núcleo que se encuentran los nucleones, protones y neutrones, partículas de dimensiones también en el orden del fm y con carga positiva y neutra respectivamente. El electrón es una partícula más pequeña que los nucleones y con carga negativa, tiene muchas características únicas, pero no se detallará a profundidad aquí. Es del interés de la investigación presente, el núcleo atómico y las fuerzas que median.

La Cromodinámica Cuántica o QCD es la teoría que describe la interacción fuerte, responsable de unir quarks y gluones en hadrones como protones y neutrones. QCD es una teoría de campo cuántico que se basa en el concepto de una teoría de gauge no abeliana con simetría de grupo SU(3). A diferencia del electromagnetismo, que se describe mediante la teoría de gauge abeliana U(1) y tiene al fotón como su bosón mediador, la QCD involucra un conjunto de ocho gluones que actúan como mediadores de la interacción fuerte.

El grupo de simetría SU(3) de la QCD describe cómo los quarks transforman sus “colores” (cargas de interacción fuerte). Los quarks vienen en tres “colores”: rojo, verde y azul, mientras que los gluones transportan una combinación de color y anti-color, lo que les permite interactuar tanto con quarks como entre sí. Esta auto-interacción de los gluones, una característica distintiva de las teorías no abelianas, es responsable de propiedades exclusivas de la QCD como el **confinamiento** y la **libertad asintótica**.

Confinamiento y libertad asintótica

Una de las principales propiedades de la teoría QCD es el confinamiento, que implica que los quarks y gluones no pueden existir de forma aislada en la naturaleza, sino que están siempre confinados dentro de los hadrones. Esto se debe a que el potencial entre quarks aumenta linealmente con la distancia y por tanto la fuerza es constante y supera otras fuerzas mientras esté en el rango efectivo, lo que contrasta con la fuerza electromagnética, que disminuye al separarse las cargas. Matemáticamente, el confinamiento se refleja en el potencial QCD,

⁵Una comparación habitual es la de poner una arveja o guisante en la mitad de un estadio, diciendo que esta sería el núcleo, y el estadio el átomo.

que es aproximadamente lineal a grandes distancias $V(r) \propto r$. Por otro lado, a distancias extremadamente pequeñas (o altas energías), los quarks y gluones se comportan casi como partículas libres, un fenómeno conocido como **libertad asintótica**⁶. En términos matemáticos, la constante de acoplamiento fuerte $\alpha_s(Q^2)$ disminuye logarítmicamente con el aumento del momento transferido, lo que explica por qué la interacción fuerte se vuelve débil en colisiones de alta energía.

La teoría QCD, propone que protones y neutrones son atraídos por una “fuerza fuerte” que contrarresta la fuerza electrostática que debería repeler a los protones. Esta fuerza fuerte tiene como principal mediador al gluón, una partícula elemental tipo bosónica, y como principal actor al quark, la partícula subatómica fundamental de la materia, fermiónica en naturaleza, que constituye otras partículas llamadas hadrones, una familia de partículas subatómicas.

Dentro de los hadrones se encuentran los bariones, compuestos comúnmente de tres quarks, y mesones, que se componen de un quark y un anti-quark. Ejemplos de bariones son protones y neutrones, mientras que ejemplos de mesones son piones y kaones. La mayoría de hadrones⁷, son inestables y decaen a otras partículas más pequeñas.

A pesar de su éxito en describir la interacción fuerte, la Cromodinámica Cuántica (QCD) aún presenta desafíos fundamentales y se dice que es *incompleta* mientras no sepamos hacer los cálculos. Una de las limitaciones más notables es que, aunque QCD predice el confinamiento de quarks y gluones dentro de hadrones, no existe una demostración analítica rigurosa de esta propiedad a partir de los principios básicos de la teoría. Este problema se conoce como la **hipótesis de confinamiento**, y su solución formal sigue siendo uno de los grandes problemas abiertos en física teórica. Además, QCD en el régimen de bajas energías es altamente no perturbativo, lo que significa que las técnicas de teoría de perturbaciones (basadas en series de expansión) no son aplicables. Como resultado, el estudio de los estados ligados y el cálculo preciso de

⁶Esta propiedad fue descubierta por David Gross, Frank Wilczek y David Politzer, quienes recibieron el Premio Nobel en 2004. [4][5]

⁷Hadrones como el neutrón y el protón, son los hadrones más estables. El neutrón tiene vida media de alrededor de 15 minutos, y el protón de al menos 10^{34} años, aunque perfectamente puede que nunca decaiga. Como el universo lleva 10^{10} años, queda mucho tiempo para ver si un protón decae e invalide muchas teorías.

observables hadrónicos requieren métodos numéricos como *Lattice QCD*, que, aunque poderosa, es computacionalmente intensiva y limitada en cuanto a resolución y tamaño de espacio-tiempo. Esta incompletitud refleja la necesidad de nuevas herramientas teóricas y métodos para una comprensión más profunda de la física de los hadrones.

1.1.2 Semi Inclusive Deep Inelastic Scattering

El fenómeno de scattering corresponde a una amplia gamma de procesos físicos donde una partícula se desvía de una trayectoria rectilínea. Para nuestro caso, se entenderá scattering como un resultado físico de la colisión partícula-partícula, que se le llamará **evento** o evento de scattering. Este evento ocurrirá en una región del espacio definida y una ventana de tiempo corta, y se deberá a la interacción fundamental entre partículas subatómicas. En el caso de que no se aniquilen, se dirá que la partícula hizo scattering y podrá producir partículas o bien no generar ninguna, dependiendo de los parámetros iniciales de la partícula incidente.

El experimento consiste en la *semi inclusiva dispersión inelástica profunda* (**Semi Inclusive Deep Inelastic Scattering** o *SIDIS*), un proceso en el cual un electrón (o el caso más general un leptón) de alta energía incide en un nucleón, transfiriendo suficiente energía como para que un fotón virtual “entre” o sondee en su estructura interna, produciendo hadrones que pueden ser observados. Este proceso es clave para estudiar la estructura nuclear a nivel de quarks y gluones.

En nuestro caso, el siguiente diagrama ilustra el proceso.

Tenemos un electrón que a alta energía interactúa con un nucleón, ya sea protón o neutrón. El fotón virtual debe tener la suficiente energía de lo contrario el scattering no transferirá la suficiente energía al hadrón y será considerado como un scattering elástico.

$$\lambda = \frac{h}{p} \tag{1.1}$$

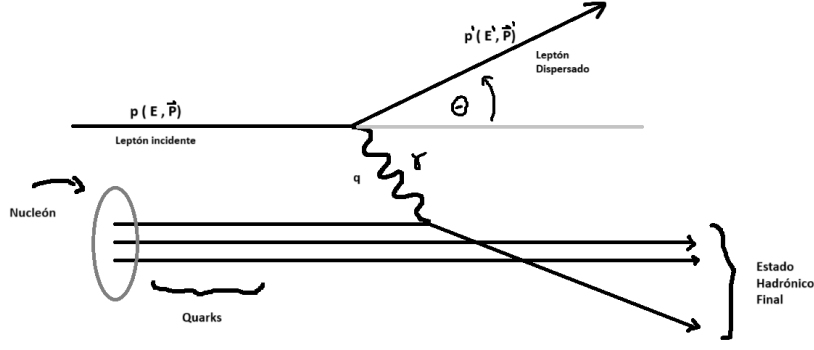


Figure 1.2: SIDIS para un leptón cualquiera impactando en un hadrón cualquiera

La cantidad invariante de Lorentz Q^2 que corresponde a la virtualidad del bosón intercambiado, o momentum transferido por el fotón. es relevante encontrar para el estudio de BEC, que se hablará adelante. Podemos encontrar Q^2 de la siguiente forma

$$\begin{aligned}
 Q^2 &\equiv -q^2 \\
 &= - \left[(E - E')^2 - (\vec{p} - \vec{p}')^2 \right] \\
 &= \left[E^2 + E'^2 - 2EE' - \vec{p}^2 - \vec{p}'^2 + 2\vec{p} \cdot \vec{p}' \right] \\
 &= 2 \left(EE' - \vec{p} \cdot \vec{p}' \right) \\
 &= 2EE' (1 - \cos(\theta)) \\
 &\approx 4EE' \sin^2\left(\frac{\theta}{2}\right)
 \end{aligned} \tag{1.2}$$

Vemos que Q^2 depende completamente del electrón, la energía inicial y final, y el ángulo en que se dispersa.

1.1.3 Bose Einstein Correlations

Las **correlaciones Bose-Einstein** (BEC) : A nivel subatómico, es imposible distinguir partículas del mismo tipo a ese nivel, se dice que son partículas idénticas indistinguibles. Podemos describir matemáticamente a una partícula como una función de onda plana.

$$\Psi_a(x_1) = e^{ik_a x_1} \tag{1.3}$$

Las partículas subatómicas que tienen un spin entero se denominan como Bosones, y tienen una función de onda compuesta simétrica y si tienen espín semi entero son Fermiones con una función de onda compuesta antisimétrica.

$$\Psi_{\pm}(x_1, x_2) = \frac{1}{\sqrt{2}} (\psi_a(x_1)\psi_b(x_2) \pm \psi_b(x_1)\psi_a(x_2)) \quad (1.4)$$

El fenómeno de correlación Bose-Einstein fue observado por primera vez en experimentos de física nuclear y de partículas a mediados del siglo XX Por G.Goldhaber en una aniquilación $\bar{p}p$ [6] [7], inspirado por el efecto Hanbury Brown y Twiss [8] en astronomía. En física de alta energía, la teoría de las BEC se ha aplicado para interpretar la producción de partículas en colisiones hadrónicas, como las que ocurren en aceleradores de partículas.

El comportamiento estadístico de los bosones es descrito por la estadística Bose Einstein. Originalmente propuesto por Albert Einstein y Satyendra Nath Bose, esta estadística describe el comportamiento colectivo de partículas idénticas que no cumplen el principio de exclusión de Pauli. Si volvemos al formal-

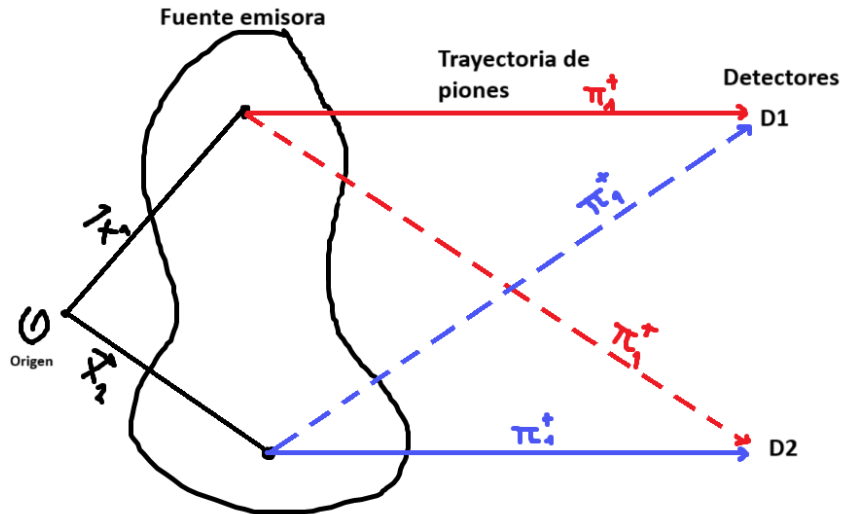


Figure 1.3: La trayectoria de dos piones positivos indistinguibles

ismo, sabemos que como característica, dos bosones indistinguibles, pueden compartir estados cuánticos y los fermiones no, por lo que se dice que Solo los fermiones siguen el **principio de exclusión de Pauli**. La función compuesta de dos bosones idénticos es la siguiente

$$\Psi_+(x_1, x_2) = \frac{1}{\sqrt{2}} (\psi_a(x_1)\psi_b(x_2) + \psi_b(x_1)\psi_a(x_2)) \quad (1.5)$$

A continuación tomamos la función de onda compuesta y elevamos al cuadrado desarrollando de la siguiente forma

$$\begin{aligned} |\Psi_+(x_1, x_2)|^2 &= \frac{1}{2} [(e^{ik_a x_1} e^{ik_b x_2} + e^{ik_b x_1} e^{ik_a x_2})(e^{-ik_a x_1} e^{-ik_b x_2} + e^{-ik_b x_1} e^{-ik_a x_2})] \\ |\Psi_+(x_1, x_2)|^2 &= 1 + \frac{1}{2} [e^{i(k_a - k_b)x_1} e^{i(k_b - k_a)x_2} + e^{i(k_b - k_a)x_1} e^{i(k_a - k_b)x_2}] \end{aligned}$$

Si hacemos la sustitución $q \equiv k_a - k_b$

$$\begin{aligned} |\Psi_+(x_1, x_2)|^2 &= 1 + \frac{1}{2} (e^{iqx_1} e^{-iqx_2} + e^{-iqx_1} e^{iqx_2}) \\ |\Psi_+(x_1, x_2)|^2 &= 1 + \frac{1}{2} (e^{iq(x_1 - x_2)} + e^{-iq(x_1 - x_2)}) \\ |\Psi_+(x_1, x_2)|^2 &= 1 + \text{Cos}(q(x_1 - x_2)) \end{aligned} \quad (1.6)$$

Con esto ahora sabemos que la densidad de probabilidad depende de $\mathbf{q}$ que corresponde a la diferencia de momentum y de las posiciones de las fuentes.

Sin embargo este análisis es simplificado, pues considera que hay 2 fuentes de bosones. Una generalización[9] tiene la siguiente forma:

$$\begin{aligned} P_{(k_a, k_b)} &\propto \int d^3 r_1 d^3 r_2 \rho(r_1) \rho(r_2) |\Psi_+(x_1, x_2)|^2 \\ P_{(k_a, k_b)} &= 1 + |\tilde{\rho}(q)|^2 \end{aligned} \quad (1.7)$$

Con la transformada de Fourier de $\tilde{\rho}(\mathbf{r})$ La función de correlación es entonces

$$R(T) = 1 + |\tilde{\rho}(T)|^2 \quad (1.8)$$

con $T \equiv \sqrt{-(p_1 - p_2)^2}$.⁸

⁸T corresponde a la diferencia de momentum, una cantidad invariante de Lorentz que es la que se usa en la función de correlación. También se le conoce como Q_{12} y no se debe confundir con la variable Q que representa la transferencia de momentum. Por ese motivo, se denotará como $\mathbf{T}$ Para evitar confusiones

Si consideramos que pueden haber fuentes coherentes e incoherentes surgiendo de la Fuente, tendremos la siguiente ecuación

$$R(T) = 1 + \lambda |\tilde{\rho}(T)|^2 \quad (1.9)$$

Donde decimos que lambda es un factor empírico que corresponde al grado de incoherencia de la Fuente, “0” es una fuente coherente, y “1” una fuente completamente incoherente. Si implementamos algo conocido como la parametrización de Goldhaber, decimos que la fuente donde se emiten los piones es una gaussiana del tipo

$$\rho(r) = \rho(0) e^{-\frac{r^2}{2R^2}} \quad (1.10)$$

La ecuación 1.9) o función BEC tendrá la siguiente forma.

$$R(T) = 1 + \lambda e^{-r_G^2 T^2} \quad (1.11)$$

La ec 1.11) es conocida como la parametrización de Goldhaber.

Podemos generalizar aun más la expresión, agregando un factor gamma de normalización y un polinomio para describir las correlaciones a larga distancia.

$$R(T) = \gamma \cdot [1 + \lambda e^{-r_G^2 T^2}] \cdot P(T) \quad (1.12)$$

Esta será la ecuación teórica de la función de correlación que se usara más tarde, donde γ corresponde a un factor de normalización, λ es el parámetro de caoticidad, r o r_G es el radio de la fuente de emisión de piones y el polinomio $P(T)$ puede ser de cualquier grado, pero preferentemente se usará de grado 2, por motivos del ajuste.

1.1.4 Estudios Previos

Uno de los primeros experimentos en donde se mide Multiplicity Ratio en experimentos con dos objetivos o *targets* distintos, fue el de *Stanford Linear Accelerator Center* (SLAC) en 1978.[10]. Desde entonces diversos experimentos fueron dirigidos por diversas colaboraciones, con *targets* variados.

La colaboración *BBCN (Big Bubble Chamber Neutrino)* en 1993 usó la parametrización de Goldhaber para encontrar el radio de emisión de piones y el parámetro de caoticidad empírico λ que serán nombrados en la discusión.[11] Esto se logró en la cámara *15 Foot Bubble Chamber* estudiando dos piones del mismo signo a partir de una colisión entre un neutrino y antineutrino. No encontraron diferencias sustanciales entre *targets* distintos en cuanto al radio de emisión, o dependencia cinemática.

La colaboración ZEUS en HERA, investigó BEC en el año 2003, usando dispersión inelástica profunda *ep* obteniendo una luminosidad integrada de 128 pb^{-1} , encontrando que las correlaciones son independientes de la virtualidad intercambiada del fotón Q^2 en el rango $1 \text{ GeV}^2 < 8000 \text{ GeV}^2$ sin encontrar diferencias significativas en el marco de Breit. Sus resultados muestran que la interferencia de Bose Einstein y por tanto el tamaño la fuente de emisión de partículas es independiente del proceso duro [12]

La colaboración Hermes también dio resultados buscando BEC, en el año 2015 , haciendo un experimento DIS e^+e^- , con *targets* desde el H_1 hasta Xe. Complementando los datos con simulaciones se encontraron BEC en todos sin una variación significativa dependiente de la masa. También no hay evidencia de dependencia de la masa invariante W. [13]

En el CERN, experimentos como ALICE o LHCb, también investigan BEC, con energías muy altas, entre 2.76 hasta 7 TeV y con luminosidades integradas de 1.06 nb^{-1} . La colaboración ALICE estudia BEC de multipiones, es decir dos tres o incluso 4 piones formando una correlación, con colisiones entre protones y núcleos de plomo. Se encuentra que hay supresión de las correlaciones de múltiples piones comparado a la de dos.[14] [15] [16].

En JLab desde 9 de enero a 5 de marzo del año 2004 dos experimentos llamados “E02-104” “E02-110” en un periodo conocido como EG-2 en la sala experimental *Hall-B* de *Thomas Jefferson National Accelerator Facility*, Virginia, EEUU. El haz de electrones proveniente de **CEBAF** *Continuous Electron Beam Accelerator Facility* simultáneamente se repartía entre tres salas experimentales *Hall-A*, *Hall-B* y *Hall-C* donde varios equipos investigaban física de altas energías. El detector de *Hall-B* se llamaba **CLAS** cuya sigla en inglés es *CEBAF*

Large Acceptance Spectrometer diseñado para funcionar con haces de electrones de 5 GeV de energía y 8 nA de corriente.[17]

1.2 Experimento RG-E y CLAS12

La sigla *CEBAF*, y corresponde a la facultad del acelerador de partículas de **Jefferson Lab** donde muchos experimentos son llevados a cabo, con el fin de investigar física de partículas de altas energías, en particular QCD. En *CEBAF*, se implementan múltiples haces de electrones de varios GeV con gran precisión e intensidad siendo el rayo usado de 12 GeV en la actualidad, mejorado de 6 GeV con respecto al experimento anterior EG-2. El rayo del acelerador *CEBAF* llega a 4 salas experimentales o “Hall”, donde se investigan temas distintos a partir del scattering del electrón. En particular Hall B se encuentra el **CEBAF Large Acceptance Spectrometer** para las operaciones de haces de electrones 12 GeV, para estudiar reacciones nuclear y hadrónicas electro-inducidas.[18]. CLAS 12 tiene muchos detectores con funciones distintas para la identificación de partículas como Contadores de Cherenkov, centelleadores *time-of-flight*, y calorímetros electromagnéticos.

El experimento RG-E consiste en una colisión electrón protón a altas energías con el espectrómetro CLAS12, con un haz de electrones de 12 GeV, y es una extensión del trabajo realizado previamente con CLAS con energías de 5 GeV.[17] Este busca con las nuevas tecnologías en sensores y mayor energía del haz, obtener datos más refinados y abundantes para diversas observaciones. El experimento tiene como principal característica la introducción de un sistema de dos blancos, objetivos o **target**, con el fin de analizar como la colisión se desarrolla y los resultados que deja en dos medios distintos, pero con el mismo haz de electrones.[19] El electrón a colisionar, primero incide en un medio de **Deuterio líquido**, y luego un medio **sólido**, que puede ser **Carbono, Aluminio, Cobre, Estaño o Plomo**. En las figuras 1.4) y 1.5) se muestran el diagrama de *double target* y una foto del sistema.

Las imágenes de las figuras 1.4) y 1.5) fueron proporcionadas por el profesor Hayk Hakobyan desde fuentes oficiales de JLab y el experimento RG-E.

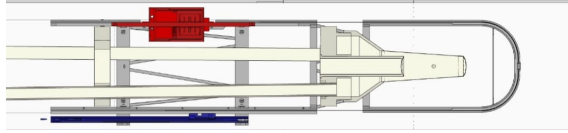


Figure 1.4: Diseño sistema double target usado en CLAS12

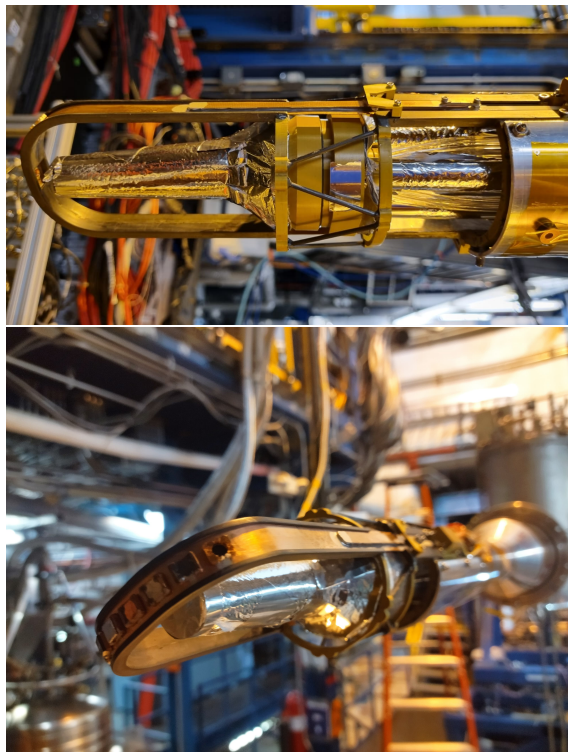


Figure 1.5: Sistema de Double Target

2

Método

En esta sección de la investigación se detallará el como se obtuvieron los resultados del experimento a partir de los datos entregados.

Comenzaré explicando la herramienta principal de la investigación, ROOT[20]. Luego se detallará el proceso paso a paso explicando con profundidad los conceptos relevantes

2.1 Conceptos ROOT y c++

Para el análisis y procesamiento de los datos proporcionados, se utilizó ROOT, un framework orientado a objetos en C++ desarrollado en CERN, diseñado para el almacenamiento, procesamiento, visualización y análisis de datos científicos. Las características principales son la siguientes:

- **Almacenamiento:** Los datos y objetos de C++ se almacenan en archivos ROOT (file.root) en un formato binario comprimido y autodescriptivo. ROOT ofrece la estructura de datos TTree, optimizada para el análisis estadístico de grandes conjuntos de datos. Un TTree contiene múltiples ramas (TBranch), que a su vez contienen hojas (TLeaf). Las hojas suelen representar variables físicas, como energía o momento. Esta estructura permite un acceso eficiente y organizado a los datos.
- **Procesamiento:** Los archivos ROOT pueden ser accedidos de manera eficiente gracias a la estructura de TTree y al manejo avanzado de memoria que proporciona ROOT. Además, ROOT ofrece una amplia gama de herramientas matemáticas y estadísticas para manipular y analizar datos, incluyendo clases de álgebra lineal, algoritmos numéricos y métodos de análisis de regresión.
- **Visualización:** ROOT incluye herramientas avanzadas para la creación de gráficos de datos científicos. Los datos pueden representarse mediante

histogramas, gráficos bidimensionales y tridimensionales, e incluso visualizaciones interactivas. Con estas opciones, se pueden interpretar fácilmente patrones y relaciones dentro de los datos. Adicionalmente, el usuario puede personalizar las propiedades visuales, como colores, estilos de líneas y etiquetas, para optimizar la presentación de resultados.

- **Análisis:** ROOT ofrece un conjunto robusto de herramientas para análisis estadístico, que incluye ajuste de curvas (*fitting*), análisis multivariante y reducción dimensional. Estas funcionalidades están integradas en las clases de ROOT y pueden combinarse con macros en C++ para realizar estudios complejos y detallados. Esto permite identificar características relevantes en los datos y extraer conclusiones científicas con precisión.
- **Macros en C++:** ROOT permite la creación de macros en C++ para automatizar tareas de análisis. Estas macros se desarrollan de manera interactiva utilizando el intérprete de C++ Cling, lo que facilita la ejecución y depuración en tiempo real. Una vez finalizadas, las macros pueden ejecutarse a velocidad compilada sobre grandes conjuntos de datos, ya sea mediante compilación en tiempo real o creando programas independientes para procesamiento por lotes.

2.2 Datos Preprocesados

Los datos que fueron proporcionados para esta investigación fueron preprocesados con el objetivo de simplificar el proceso. A continuación se presentará el tipo de archivo con el que se trabajó, con las variables en formato “leaf” y que representan:

Cada archivo es nombrado como “ntuples_dc_020XXX.root” donde se almacenan los datos del experimento, para la sesión o **run 20XXX**, denotada como hacer “correr” el experimento de manera continua guardando los datos generados en esa sesión. Cada sesión, tiene exactamente un *target* solido y uno liquido, donde el solido cambia y el liquido siempre corresponde a Deuterio. Para esta investigación se proporcionaron 15 archivos cada uno con una *run*. Se mostraran en orden ascendiente de numero corresponde a los siguientes target.

Table 2.1: En la tabla se presentan los archivos usados, con sus numeros de *run* y los *target* asociados.

Run number (020 + ____)	Targets
82	LD2 + Pb
136	LD2 + C
150	LD2 + C
176	LD2 + C
192	LD2 + Cu
201	LD2 + Cu
226	LD2 + Cu
267	LD2 + Pb
317	LD2 + Pb
339	LD2 + Sn
392	LD2 + Sn
417	LD2 + Sn
439	LD2 + Al
476	LD2 + Al
486	LD2 + Al

Cada archivo tiene una cantidad de **entradas o “entries”** entre el orden de 10^5 y 10^6 . Cada entry representa a una partícula y sus cantidades medidas en los detectores se representan en las **variables** en formato “*leaf*”. Hay 38 variables que permiten identificar las partículas y sus cantidades respectivas que permitan cálculos posteriores. Con asterisco se marcaran las relevantes para esta tesina.

- 1) **run_num***: Corresponde al numero de *run* o *session* de la entrada. Es un numero entero del tipo 20XXX. Sirve para identificar las partículas y distinguir sus números de eventos cuando se fusionan archivos Ej : run_num = 20082
- 2) **evnt_num***: corresponde al numero de evento de la entrada. Es un numero entero que va desde 1 hasta un numero del orden 10^5 a 10^6 dependiente del archivo. Más de una entrada pueden compartir el mismo numero de evento en un mismo archivo y esto corresponde a varias partículas provenientes del mismo evento o colisión. ¹EJ: evnt_num = 253403
- 3) **E_beam** Corresponde a la energía medida del rayo.
- 4) **pid***: Corresponde al tipo de partícula medida. Es un numero entero y se puede identificar el tipo de partícula mediante una tabla. Vale la pena mencionar que esta identificación es parte del preprocesamiento y puede ser identificado con otras variables, como carga o masa. En nuestro caso usamos pid=11 y pid=211 para electrón y pion positivo respectivamente.

¹Recordar que un evento corresponde a una colisión (refiérase a 1.1.2))

- 5) **charge**: corresponde al valor de la carga de la partícula. Puede tener dos valores, -1 o 1. Ej: charge=1
- 6) **status**: Variable relacionada al rol de la partícula detectada.
- 7) **mass**: corresponde a la masa de la partícula medida. Tiene un valor *float* y está en unidades de GeV/c^2 . Ej: “masa $\pi^+ \rightarrow$ mass = 0.13957018”
- 8) **v_x**: corresponde al componente en el eje x de la posición de vértice reconstruido cuando ocurre el evento o *scattering*.(float)
- 9) **v_y**: corresponde al componente en el eje y de la posición de vértice reconstruido cuando ocurre el evento o *scattering*.
- 10) **v_z***: Corresponde al componente en el eje z de la posición de vértice reconstruido cuando ocurre el evento o *scattering*. Es particular útil para distinguir el tipo de *target*
- 11) **p_x***: Corresponde al componente en el eje x del momentum de la partícula en el Marco de referencia de laboratorio (Lab frame) (float)
- 12) **p_y***: Corresponde al componente en el eje y del momentum de la partícula en el Lab frame
- 13) **p_z***: Corresponde al componente en el eje z del momentum de la partícula en Lab frame
- 14) **p***: Magnitud del vector del momentum. Todas estas cantidades son float.
- 15) **theta***: Ángulo polar de la partícula relativo al eje del rayo o eje z.
- 16) **phi***: Ángulo azimutal de la partícula en Lab frame
- 17) **beta**: La velocidad medida por el detector en relación a la velocidad de la luz. $\beta = \frac{v}{c}$
- 18) **trigger_status**: Indica que detector activo el evento.
- 19) **sector**: Corresponde a la región geométrica en la que se midió la partícula.
- 20) **chi2**: Un parámetro de calidad del fit de la partícula en el detector.
- 21) **NDF**: Number of Degrees of Freedom representa los grados de libertad en el proceso de track fit . Se usa para calcular
- 22) **E_PCAL**: Energía depositada por la partícula en la región del calorímetro pre shower
- 23) **E_ECIN**: Energía depositada por la partícula en la región del calorímetro interno

- 24) **E_ECOU**: Energía depositada por la partícula en la región del calorímetro externo
- 25) **E_TOTAL**: Energía total depositada en los calorímetros.
- 26) **Delta_TOF**: La diferencia de tiempo de vuelo, usada para identificar partículas.
- 27) **NPhe_LTCC**: *Number of photoelectrons in Low Threshold Cherenkov Counters* usado para identificar partículas
- 28) **NPhe_HTCC**: *Number of photoelectrons in High Threshold Cherenkov Counters* usado para identificar partículas
- 29) **Q2***: Transferencia de cuadrimomentum al cuadrado. es decir la diferencia entre momentum del electrón incidente y dispersado. Representa la resolución del fotón virtual, o la virtualidad del fotón.
- 30) **nu**: Transferencia de energía del electrón al fotón
- 31) **x_b**: X Bjorken, $x_b = \frac{Q^2}{2M\nu}$ con M la masa del protón. Representa la fracción de momentum del partón golpeado
- 32) **y_b***: Y Bjorken, $y_b = \frac{p \cdot q}{p \cdot l} = \frac{E_l - E'_l}{E_l}$. Representa la inelasticidad
- 33) **W2***: La masa invariante del sistema hadrónico describe la masa del estado intermedio
- 34) **z_h**: Fracción de la energía del fotón virtual transferida al hadrón.
- 35) **p_T2**: Momentum transversal al cuadrado. Útil para obtener BEC en dos dimensiones
- 36) **p_L2**: Momentum longitudinal al cuadrado.
- 37) **phi_PQ***: Ángulo azimutal entre el plano de scattering y el plano de producción del hadrón marco centro masa entre el protón y el fotón virtual:
- 38) **theta_PQ***: Ángulo polar entre el hadrón y el fotón virtual en el mismo marco de referencia.

Las variables importantes para el procesamiento de datos son:

run_num, **evnt_num**, **pid**, **v_z**, los componentes de **p** y la variable **p** en sí, los ángulos **theta** y **phi**. También **phi_PQ** y **theta_PQ** para una corrección a la distribución *background*. Las variables

Q2, y_b, W2 serviran para hacer cortes *DIS*.^{2 3}

2.3 Procesamiento de datos

Ya conociendo los datos comenzaremos a obtener las correlaciones Bose Einstein. Para eso tenemos que tener en cuenta las ecuaciones de la introducción. La función de correlación teórica está dada por 1.12), La función de correlación experimental está dada por

$$R(T) = D_{signal}(T)/D_{background}(T) \quad (2.1)$$

Donde D(T) corresponde a la distribución de las entradas con respecto a la variable T, para cada par.

El objetivo es encontrar las distribuciones **signal**, que usa piones dentro del mismo evento y por tanto **correlacionados**, y calcula su T, y luego la distribución **background**, que usa piones **no correlacionados**. Con estas dos distribuciones se encuentra la función de correlación R(T) y se le hace un ajuste de curva según la función de correlación teórica.

2.3.1 Cortes/cuts

Para poder clasificar los datos y poder manejar la información correctamente, debemos procesar los datos por lo siguientes criterios.⁴

Particula

Usamos la variable **pid**

- Pion positivo: **pid==211**
Para el calculo de las correlaciones Bose-Einstein usaremos el pion positivo.
- Electron: **pid=11** Para cálculos varios usaremos el electrón. Estos cálculos incluyen, la obtención de las zonas de aceptación de los targets usando **v_z**.

²Si se quisiese hacer un análisis bidimensional, las variables **p_T2** y **p_L2** se usarían, pero no es el foco de esta investigación

³La variable **v_z** es distinta en cada **sector** representado por la variable del mismo nombre, pero ese análisis no se hizo en este trabajo con el fin de simplificar resultados

⁴Revise A.1

Target u Objetivo

Se usa la variable v_z para distinguir entre tipos de *targets* mediante los siguientes cortes:

- Target Liquido (Deuterio)
Condiciones $-9 < v_z < -3.5$
- Target Solido (Carbono, Aluminio, Cobre, Estaño, Plomo)
Condiciones $-2.5 < v_z < 3$

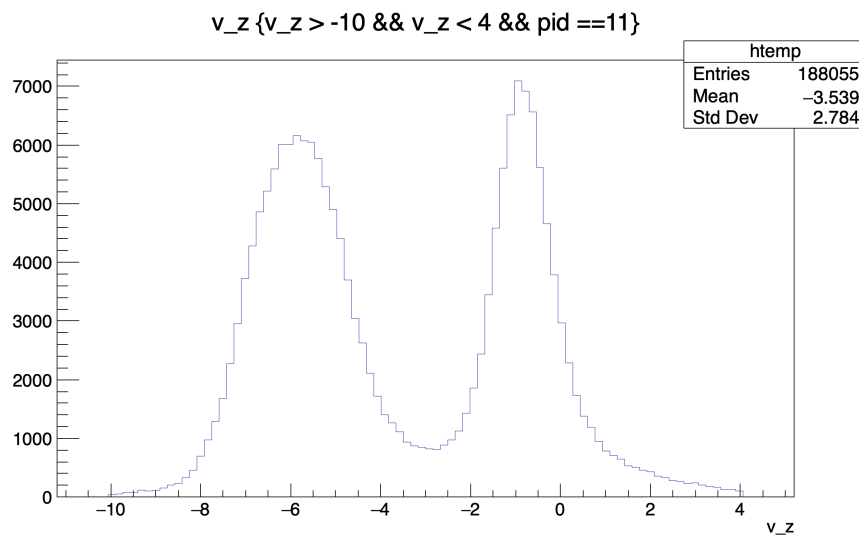


Figure 2.1: Un histograma respecto a la variable v_z para observar los target.

De la figura 2.1) podemos ver claramente como hay dos distribuciones parecidas a unas gaussianas que representan al target liquido y al solido de izquierda a derecha. Notamos que al hacer los cuts hay una zona ambigua ($-3.5 < v_z < -2.5$), donde no es fácil distinguir entre ambos target, por lo que se descarta.⁵

Por DIS, o cortes cinemáticos

Usamos las variables Q^2 , W^2 , e y_b

- $Q^2 > 1.0$:
Para excluir aquellas interacciones que no sean suficientemente profundas como para sondear dentro del protón.
- $W^2 > 4.0$:
Para excluir hadrones provenientes del decay de resonancia nuclear (como ρ y otras partículas pesadas)

⁵Aun los datos pueden calibrarse más y es de esperar que estas distribuciones de v_z se distingan aun más en un futuro.

- $y_b < 0.85$:

Para excluir fuentes de contaminación como efectos radioactivos

Estos cortes se ven visualizados en las figuras 2.2) contrastada con 2.3

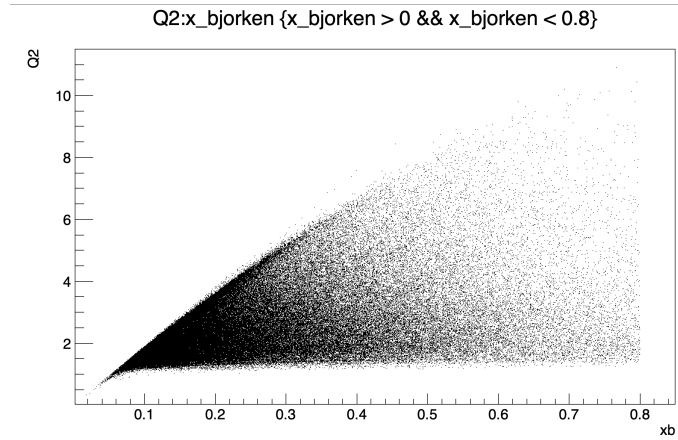


Figure 2.2: Una distribución Q2 sobre x_b del archivo ntuples_dc_020082.root

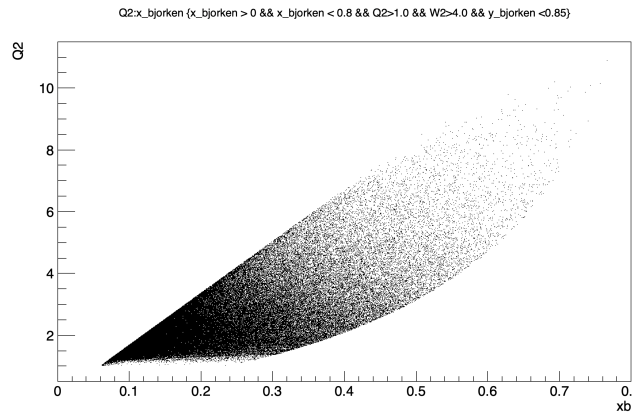


Figure 2.3: Los cortes DIS aplicados de la figura 2.2

2.3.2 Pares de piones

Para obtener las distribuciones **Signal** y **Background** debemos agrupar los piones en pares de mismo signo.

Ahora se detallará el algoritmo para procesar los datos según los cuts, para entregar piones por evento y así armar pares. Las macros y códigos en c++ están como anexo al final.

Para facilitar los cálculos introduciremos las limitaciones o suposiciones que se implementaran usando las variables.

- En cada evento habrá solo habrán **piones positivos**. Cualquier entrada que corresponda a otra partícula será descartada.

- En cada evento habrán exactamente dos piones. Eventos que contengan 1 pion, o más de dos piones serán descartados.
- Las runs serán fusionadas en un solo archivo de nombre del tipo de target, para tener más muestras.⁶

1) Algoritmo

Consideremos las variables `run_num` y `evnt_num`, como el numero de run y el numero de evento respectivamente.

Si sabemos que en el archivo se hicieron los cuts relevantes, solamente quedan piones en rangos razonables. Con estos podemos checkear su numero de evento (y si es necesario su numero de run) para ordenarlos como necesitamos. Aqui el algoritmo que se debe seguir.

1. Se inicializan las variables `num_entries` como el numero total de entradas e `i` como un indice
2. Se usa un loop while para recorrer todas las entradas (`i=0:num_entries`)
 - (a) Se obtienen las variables del tree de la entrada actual
 - (b) Se inicializan variables de conteo=0 e `inicio=i`
 - (c) Se almacena `evnt_num` de la entrada actual como `evnt_actual`
 - (d) Se inicia otro loop while para recorrer todas las entradas (`i:num_entries`)
 - i. Se obtienen las variables del tree de la entrada actual
 - ii. Comparar si `evnt_num` es igual al `evnt_actual`. De ser así se corta este loop
 - iii. Se aumentan los indices de conteo e `i`
 - (e) Si el indice de conteo es igual al requerido (En nuestro caso `conteo==2`), Se almacena la entrada en un tree nuevo

Se crea un archivo con un solo par por evento, y según la suposición inicial, será de un tipo de target.⁷

2.3.3 Signal & background

A partir de los archivos con pares por evento, armaremos las distribuciones signal y background.

⁶Revisar el código usado A.2)

⁷El código usado es A.3

A) Distribución Signal

Corresponde a la distribución de pares correlacionados. Para crear esta distribución se usan los pares de piones usando la variable `evnt_num` almacenándola en una variable auxiliar `current_evnt`. Luego se almacenan las variables del primer evento, se aumenta el numero de entrada y se almacenan las variables del segundo. Para avanzar de par, se sigue recorriendo la entrada y si la variable `evnt_num` es distinto al `current_evnt`, se considera un nuevo par.⁸⁹

B) Distribución Background

Corresponde a la distribución de pares no correlacionados. Se puede obtener de varias formas siempre que sean dos piones en condiciones similares y no estén correlacionados. Métodos populares para encontrar estos pares no correlacionados, como *Method of Event Mixing pairs* (MEM) y *Method of Un-like Sign pairs* (MUS). Usaremos el **método MEM**, este consiste en calcular la variable T entre dos piones del mismo signo, que no pertenezcan al mismo evento y por lo tanto no haya correlación entre estos. Las ventajas de este método son que se puede elegir un numero arbitrario de pares no correlacionados y que se puede elegir cualquier pion dentro de los pares en el archivo. La mayor desventaja es que como no pertenecen al mismo evento, su fotón virtual no es el mismo y deben alinearse para tener un análisis correcto. El algoritmo para seleccionar esto sigue la siguiente forma¹⁰

1. Se inicializa una variable de numero de entradas del archivo `nentries`. Se inicializan también, variables de conteo de pares `pair_count` y una variable `max_pairs` que representa las veces que se iterará el proceso de calculo de pares
2. Se usa un loop while donde mientras `pair_count < max_pairs`
 - (a) Se almacenan las variables de una partícula dada por un numero aleatorio hasta `nentries`. Se hace nuevamente para otra partícula
 - (b) Se verifica si pertenecen al mismo evento y run. De ser así se comienza de nuevo desde el paso 1, sino se continua al siguiente paso.
 - (c) Se calcula con las variables entre las partículas y se almacenan las variables deseadas en este par.

⁸De haber más de 2 piones por evento, el proceso es similar, se recorre cada entrada hasta que se cambie de evento. Luego se calcula en cada combinación de pares un T distinto. El numero de pares siempre será C_2^n

⁹El código usado para este algoritmo es A.4

¹⁰El código es A.5

(d) Se aumenta en 1 la variable `pair_count`

Tanto para el calculo de la distribución signal o background se usaran las variables de los ángulos θ , y ϕ o alternativamente componentes del cuadrimomentos $\mathbf{p}_x, \mathbf{p}_y, \mathbf{p}_z, \mathbf{p}$. También se declara la masa del pion m_p equivalente a $0.13957018 \text{ GeV}/c^2$ El calculo de T sigue así:

$$\begin{aligned} E_1 &= \sqrt{|p_1|^2 + m_p^2} \\ p_{x1} &= p_1 * \sin(\theta_1) * \cos(\phi_1) \\ p_{y1} &= p_1 * \sin(\theta_1) * \sin(\phi_1) \\ p_{z1} &= p_1 * \cos(\theta_1) \end{aligned}$$

$$\begin{aligned} E_2 &= \sqrt{|p_2|^2 + m_p^2} \\ p_{x2} &= p_2 * \sin(\theta_2) * \cos(\phi_2) \\ p_{y2} &= p_2 * \sin(\theta_2) * \sin(\phi_2) \\ p_{z2} &= p_2 * \cos(\theta_2) \end{aligned}$$

Es importante comentar, que con los datos entregados, se puede usar también las componentes directamente, en vez de calcularlas en coordenadas esféricas.

$$\begin{aligned} T &= \sqrt{-q_4^2} \\ T &= \sqrt{-[\Delta E^2 - (\Delta p_x^2 + \Delta p_y^2 + \Delta p_z^2)]} \end{aligned}$$

Con $q_4 \equiv p_1 - p_2$ correspondiente a la diferencia de cuadrimomentos. Si calculamos T para cada par, en cada distribución deberíamos tener una distribución de tipo Landau, donde hay un peak y luego una cola larga.

2.3.4 Calculo Correlación

Teniendo las distribuciones *signal* y *background* calculadas, podemos obtener la ecuación de correlación experimental 2.1) con los algoritmos mencionados y especificados en el anexo.¹¹ Por lo que simplemente dividimos las distribuciones para obtener la distribución de correlación.

¹¹ El código usado para esta sección A.6

Es importante normalizar cada distribución para obtener una distribución significativa, pues la distribución *background* puede contener un número arbitrario de pares.

Una vez obtenido las funciones de correlación para cada *target*, podemos comparar este resultado experimental con el teórico, haciendo el ajuste ya mencionado. Debemos hacer un ajuste en intervalos significativos, por lo que vamos a **quitar** lo que no podamos explicar, o medir.¹² En este caso quitaremos el intervalo de T perteneciente a $[0, 0.10]$ del ajuste. La razón es el alto valor que hay en la distribución *signal*, algo que no es lo esperado.

Los motivos de esta discrepancia entre la distribución *signal* y *background* están explicados en la discusión.

¹²Puede parecer insensato quitar lo que no nos sirve, pero siempre que esté propiamente justificado y archivado es prudente

3

Análisis de Resultados

A continuación, se expondrá como resultaron las distribuciones *signal* y *background* en las figuras 3.1)3.2)3.3)3.4)3.5)3.6). En rojo se muestran las distribuciones *signal*, y en azul las distribuciones *background*, donde cada figura corresponde a un *target* distinto.

De las figuras, notamos un *peak* de la distribución *signal* alrededor de 0 en el *target* D2, y a menor grado en el resto de los target.

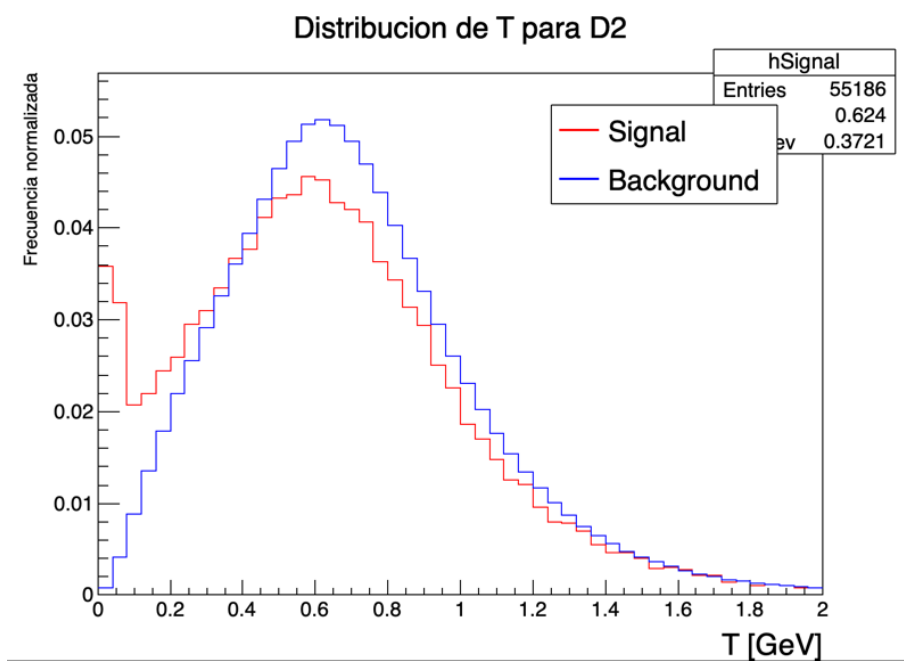


Figure 3.1: Distribuciones *signal* en rojo, *background* en azul para **Deuterio (D2)** con 50 bins

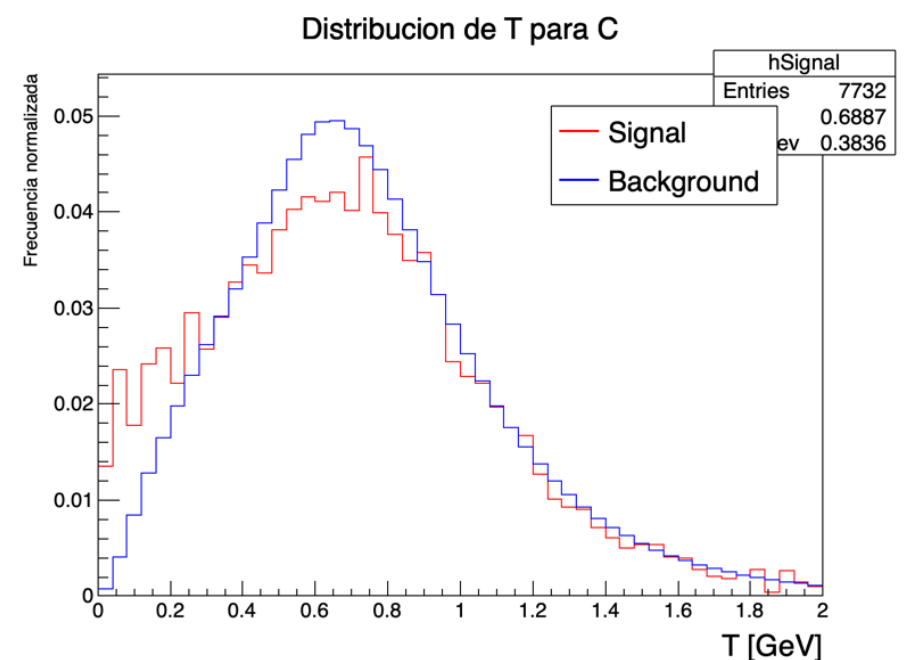


Figure 3.2: Distribuciones *signal* en rojo, *background* en azul para **Carbono (C)** con 50 bins

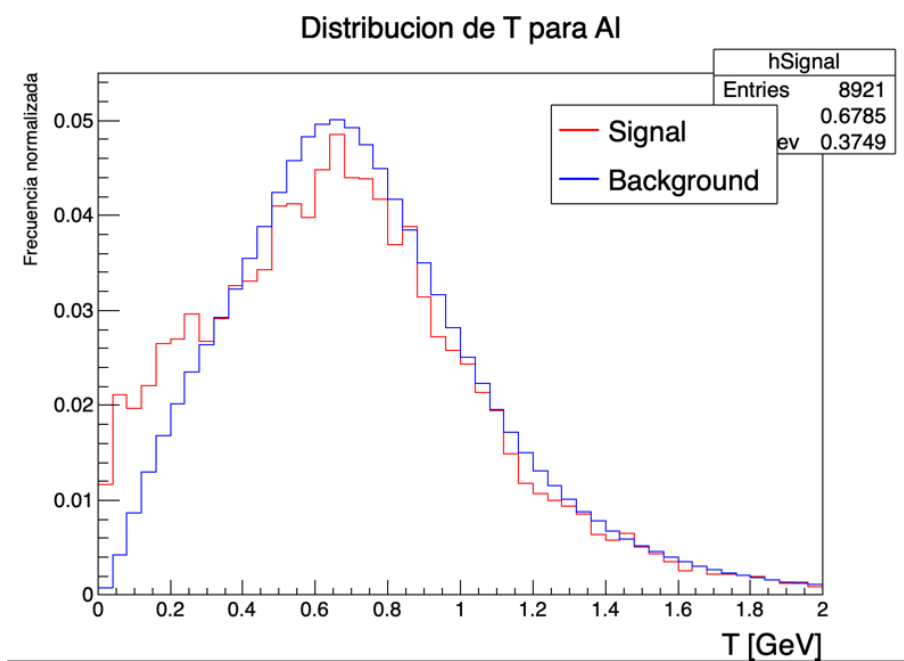


Figure 3.3: Distribuciones *signal* en rojo, *background* en azul para **Aluminio (Al)** con 50 bins

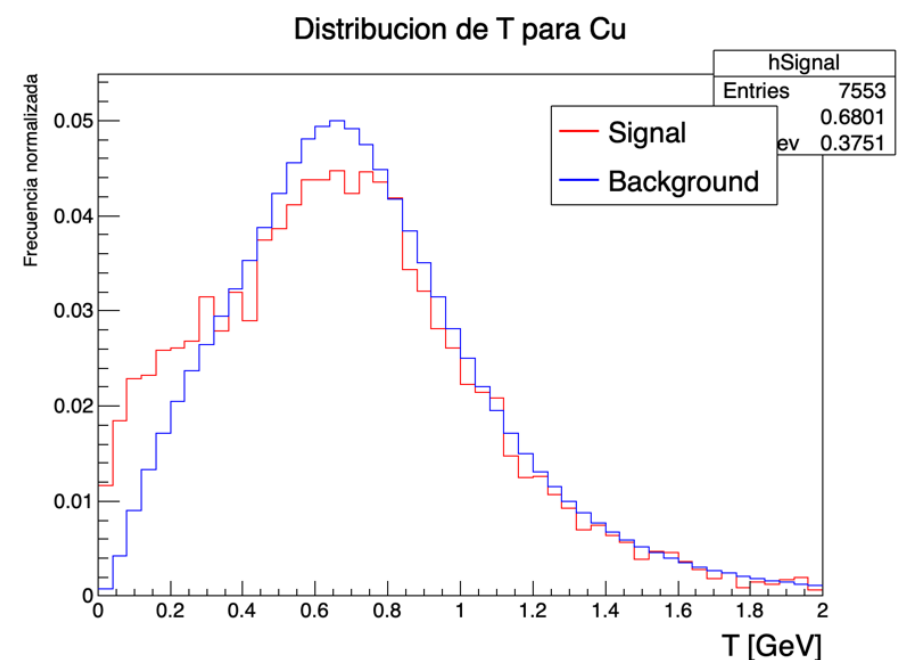


Figure 3.4: Distribuciones *signal* en rojo, *background* en azul para **Cobre (Cu)** con 50 bins

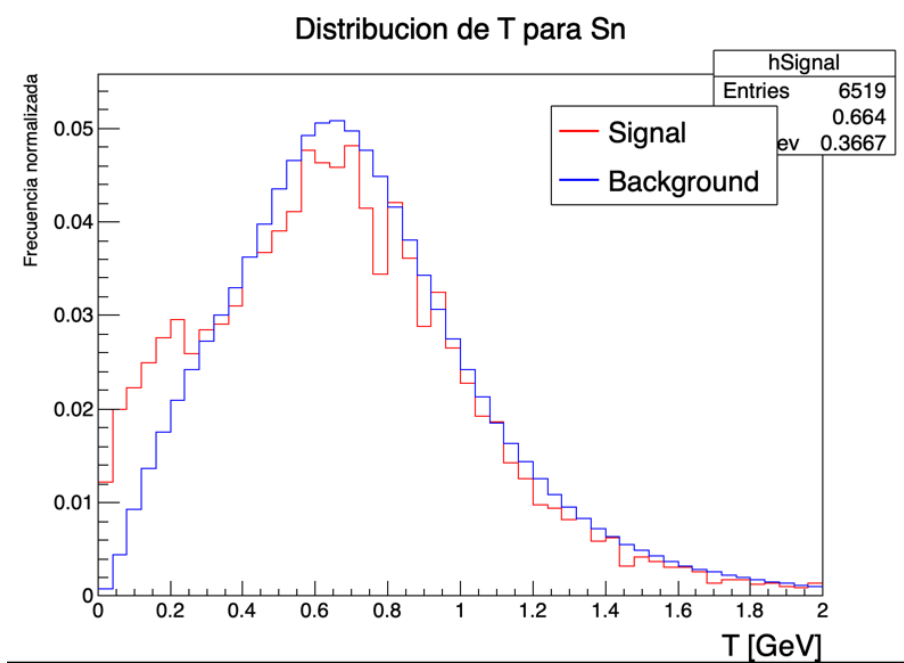


Figure 3.5: Distribuciones *signal* en rojo, *background* en azul para **Estaño (Sn)** con 50 bins

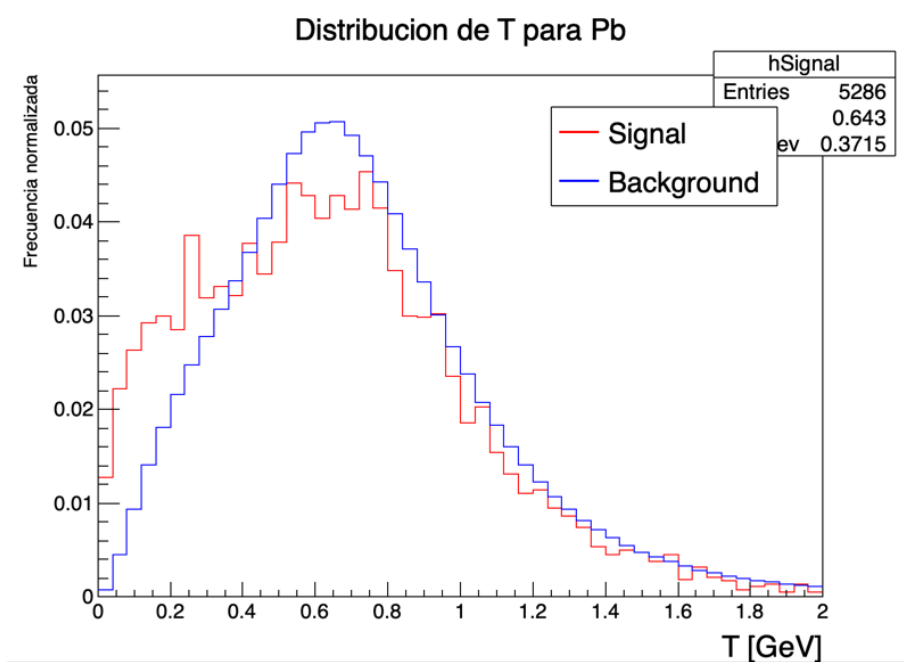


Figure 3.6: Distribuciones *signal* en rojo, *background* en azul para **Plomo (Pb)** con 50 bins

A continuación se presentan las funciones de correlación para los distintos *Target* entre 0.10 y $1.60 GeV^2$ con un número de *bins* entre 40 y 60 minimizando χ^2/NDF según la tabla 3.2). Haciendo la parametrización de Goldhaber con un polinomio de grado 2 según 1.12) se obtienen las figuras 3.7), 3.8), 3.9), 3.10), 3.11), 3.12), de Deuterio, Carbono, Aluminio Cobre, Estaño y Plomo respectivamente. Se tabulan los valores del ajuste para todos los *targets* en 3.1

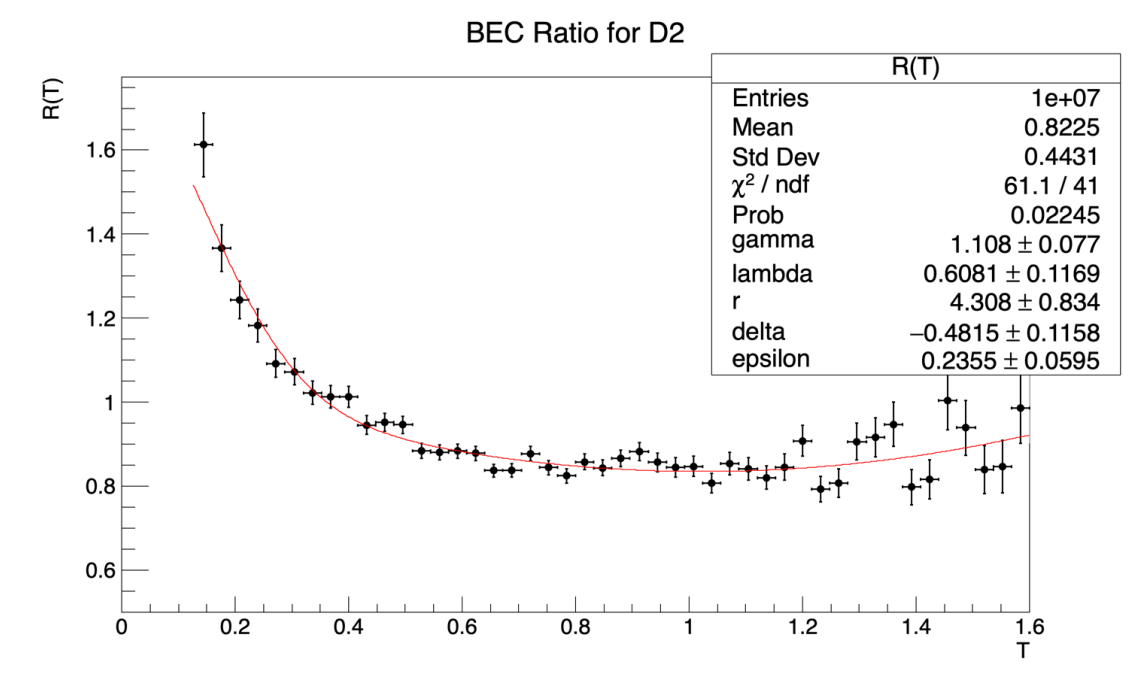


Figure 3.7: Función de correlación para el **Deuterio (D2)** con 50 bins

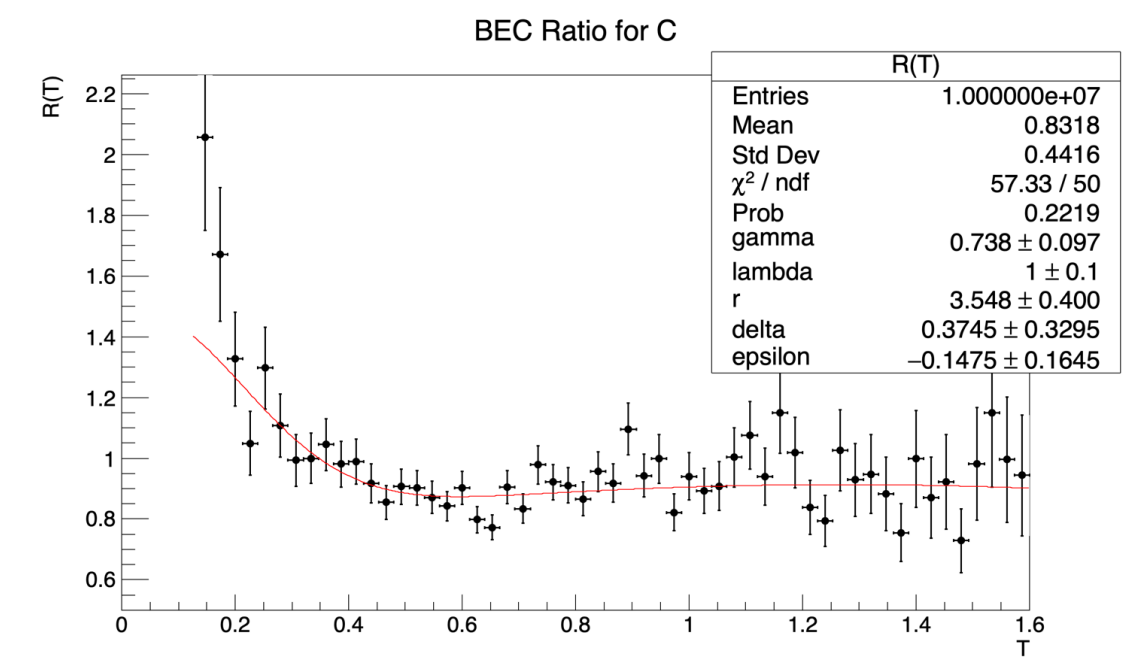


Figure 3.8: Función de correlación para el **Carbono (C)** con 60 bins

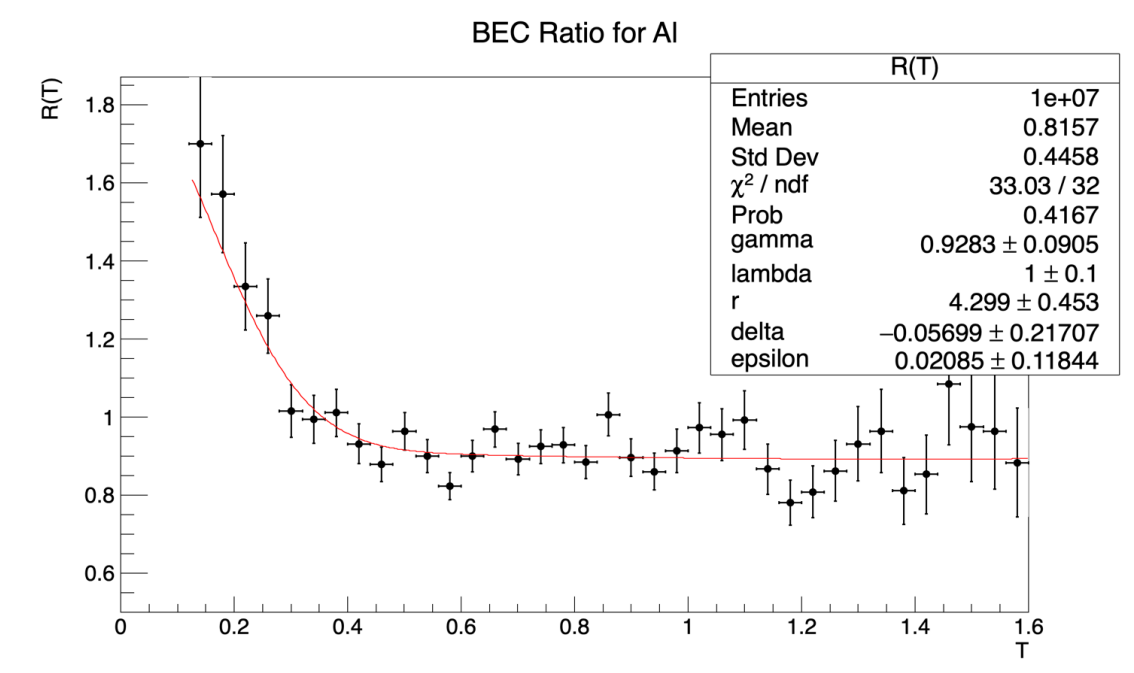


Figure 3.9: Función de correlación para el **Aluminio (Al)** con 40 bins

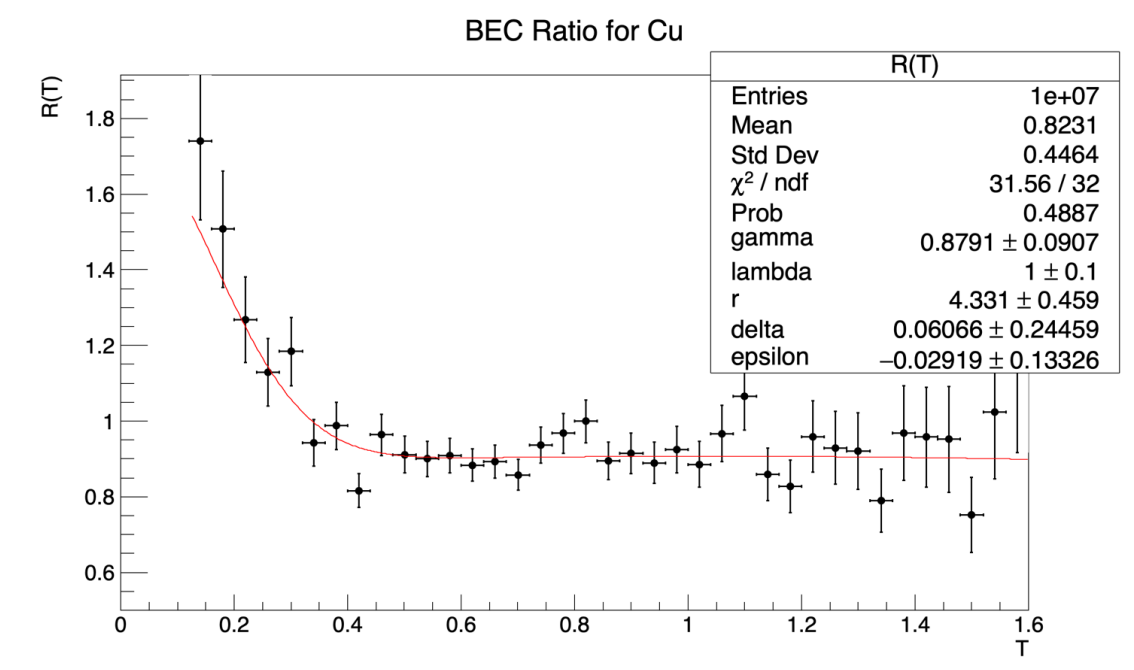


Figure 3.10: Función de correlación para el **Cobre (Cu)** con 40 bins

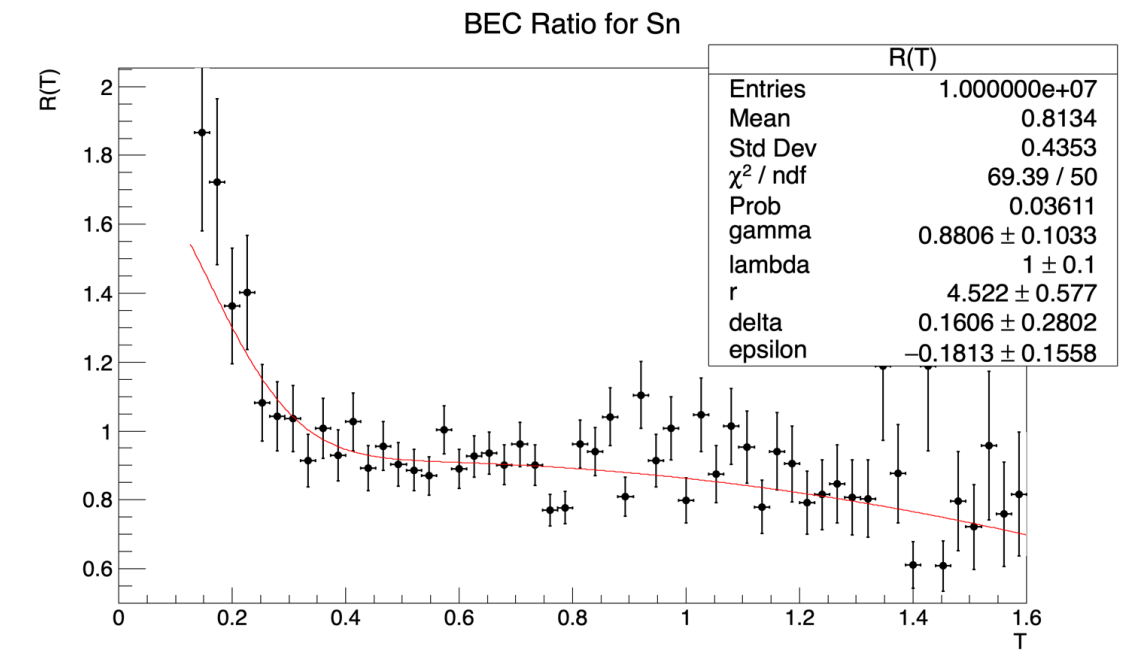


Figure 3.11: Función de correlación para el **Estaño (Sn)** con 60 bins

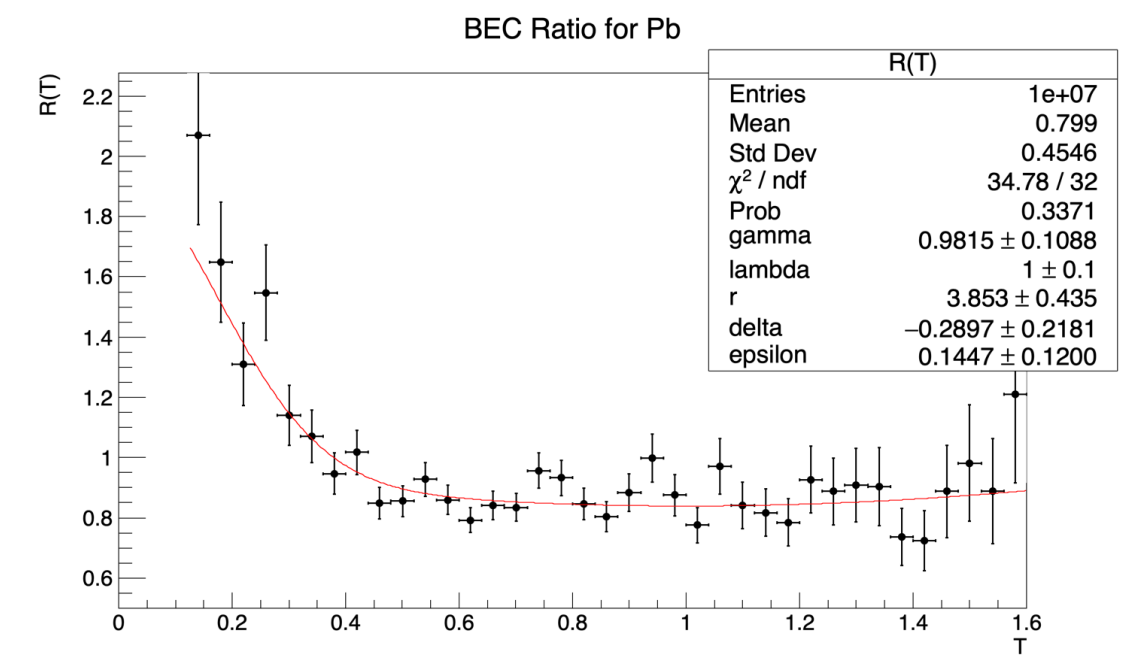


Figure 3.12: Función de correlación para el **Plomo(Pb)** con 40 bins

Los valores para todos los *targets* están tabulados en la tabla 3.1)

	D2	C	Al	Cu	Sn	Pb
#Entries	55186	7732	8921	7553	6519	5286
γ	1.108 ± 0.077	0.738 ± 0.097	0.9283 ± 0.0905	0.8791 ± 0.0907	0.8866 ± 0.1033	0.9815 ± 0.1088
λ	0.6081 ± 0.1169	$1.00 \pm 0.097^*$	$1.00 \pm 0.084^*$	$1.00 \pm 0.078^*$	$1.00 \pm 0.095^*$	$1.00 \pm 0.080^*$
r	4.308 ± 0.834	3.548 ± 0.400	4.299 ± 0.453	4.331 ± 0.459	4.522 ± 0.577	3.853 ± 0.433
δ	-0.4815 ± 0.1158	0.3745 ± 0.3295	-0.05699 ± 0.21707	0.06066 ± 0.2446	0.1606 ± 0.2802	-0.2897 ± 0.2181
ϵ	0.2355 ± 0.0505	-0.1475 ± 0.1645	0.02085 ± 0.1184	-0.02919 ± 0.13326	-0.1813 ± 0.1558	0.1447 ± 0.1200

Table 3.1: Los valores de los parámetros con el ajuste de Goldhaber. **Los valores con asterisco están limitados a 1.**

Se observa de la tabla como todos los *target* sólidos presentan una limitación en el ajuste de λ hasta 1 según limitaciones físicas según 1.1.3) Cualquier valor fuera del intervalo se podría decir que no tiene sentido físico.

La tabla 3.2) muestra los valores de χ^2/NDF para cada *target* variando la cantidad de *bins*, para $T \in [0, 1.6]$. Esta cantidad es un indicador de que tan bueno es el ajuste hecho a los datos, donde un **valor cercano a 1** es muy bueno, **menor a 1** indica *overfitting*, y **mayor a 1** indica mal *fit*.

Estadística χ^2 ; bins [20, 100]; T [0, 1.6]												
Bins	D2		C		Al		Cu		Sn		Pb	
	χ^2/NDF	=	χ^2/NDF	=	χ^2/NDF	=	χ^2/NDF	=	χ^2/NDF	=	χ^2/NDF	=
100	114.2/87	1.312	84.52/87	0.9714	89.52/87	1.0289	107.4/87	1.2344	150.1/87	1.7253	108/87	1.2517
90	106.6/78	1.362	82.36/78	1.0559	67.96/78	0.8713	77.13/78	0.9892	145.0/78	1.8705	78.03/78	1.0004
80	97.83/69	1.4178	75.45/69	1.0935	72.05/69	1.0442	79.9/69	1.1579	111.3/69	1.6130	82.04/69	1.1895
70	97.18/60	1.6196	79.51/60	1.3252	53.99/60	0.9165	75.3/60	1.255	105/60	1.75	62.53/60	1.0422
60	82.61/50	1.6522	57.33/50	1.1466	44.29/50	0.8858	44.51/50	0.8828	69.39/50	1.2052	56.86/50	1.1372
50	61.1/41	1.490	50.18/41	1.2239	37.61/41	0.9173	44.67/41	1.0895	55.23/41	1.3470	46.29/41	1.1290
40	59.17/32	1.849	38.42/32	1.2006	33.02/32	1.0321	31.56/32	0.9863	58.89/32	1.8341	34.78/32	1.0867
30	53.6/23	1.6928	26.12/23	1.1356	32.25/23	1.4021	19.24/23	0.8365	35.25/23	1.5326	30.57/23	1.3291
20	21.8/13	1.6769	19.07/13	1.4669	18.64/13	1.4338	13.64/13	1.0492	26.39/13	2.03	20.49/13	1.6146

Table 3.2: Resultados de estadística χ^2 para diferentes *bins* en el intervalo de T [0, 1.6] *GeV*.

Notamos que valores de bin entre 40 y 60 entregan los mejores resultados.

Los valores de λ y r_G corresponden a cantidades físicas y sus valores deberían estar en el intervalo $[0, 1]$ en el caso de λ y $\sim 0.4, 5$ en el caso del radio r_G . γ , δ y ϵ son parámetros libres correspondientes al factor de normalización y a parámetros del polinomio¹ y sus valores pueden tomar cualquier valor. Es posible tomar un polinomio de tercer grado, para obtener menor error, pero esto solo provocaría *overfitting*, lo que es una dirección errónea pues no entrega un resultado físico real. En la figura 3.13) se muestran los valores de λ y r_G . Las distintas co-

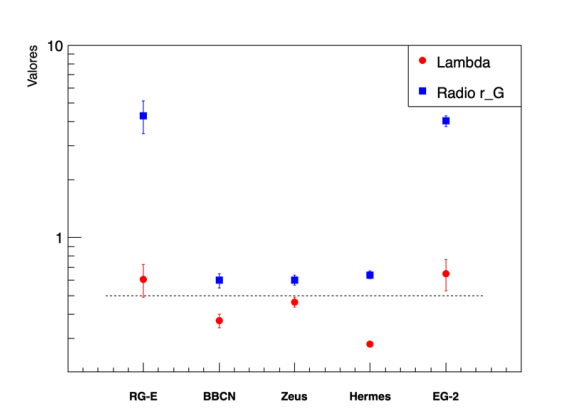


Figure 3.13: Los valores de λ y r_G para el *target* más liviano en cada una de las colaboraciones incluida EG-2 (valor por Antonio Radic) y RG-E (de esta tesina). Una línea punteada está fija en 0.5 como referencia.

laboraciones, usan colisiones distintas pero todos midiendo piones. Revise las referencias de BBCN [11]), Zeus [12]), HERMES [13])

A continuación se muestra la figura 3.14) donde se aprecian los valores de r_G para los distintos *target*.

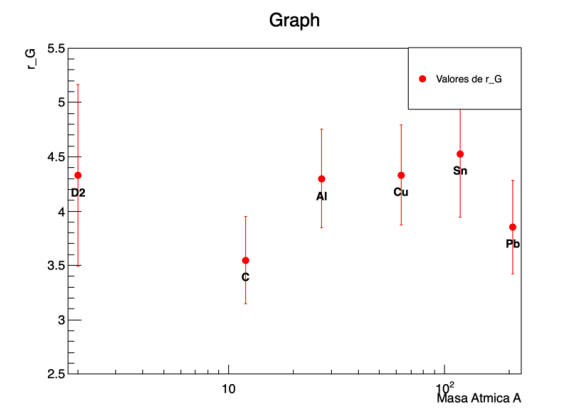


Figure 3.14: Los valores de r_G para los distintos *target*

¹ $P(T) = (1 + \delta T + \epsilon T^2)$

4

Discusión y Conclusiones

4.1 Discusión

Por el momento, los valores del ajuste de la parametrización de Goldhaber son adecuados y la función de correlación es válida. Es importante notar que la función de correlación no puede tomar valores mayores a dos por la ecuación teórica de esta, cuyo valor máximo solo puede ser 2. 1.9) Existe incertidumbre considerable que pueden ser explicado por los siguientes factores:

- La distribución *signal* que al haber tan poco espacio muestral es propenso a fluctuaciones de datos.
- Los datos aún pueden calibrarse más por lo que puede haber piones pertenecientes al otro *target*.
- Al ser datos tan preliminares, puede haber mala identificación de partículas, incluidos los piones.
- Es posible que el algoritmo de la distribución *signal* no sea del todo correcto.
- Efectos de repulsión de Coulomb entre 0.0 y 0.05.
- No hay corrección PQ, o conservación de colinealidad en la distribución *background*.

Estos resultados preliminares dan una excelente visualización cualitativa, pero la cuantitativa debe analizarse con cuidado. Podemos notar inmediatamente que a pesar del error de los *targets* sólidos, **todos los *targets* tienen una forma similar entre sí**, y los valores de r_G que oscilan entre [3.5 y 4.5] para todos los *targets* se condice con los resultados de EG-2, y el hecho que se mantenga a través de los target indica que **El radio de emisión de piones r_G no varía en cuanto a *targets*** como se observa en las colaboraciones BBCN, Zeus, Hermes.

4.1.1 Explicaciones para las incertidumbres

A continuación, se explicará a fondo los problemas que se encontraron y algunos detalles que no pudieron ser explicados.

Datos preprocesados de forma preliminar

Los datos fueron entregados preprocesados, con las partículas identificadas de forma preliminar. Es posible que hayan sido mal identificadas, pero de ser así, ha de ser minúsculo. Las variables como sector no están completamente alineadas, y la variable v_z o vértice reconstruido no tiene una resolución tan buena.

Distribución Signal

En esta distribución es donde se genera la mayor más diferencia entre otras colaboraciones, generando mucha incertidumbre como también se reporta con frecuencia. La cantidad de pares ronda en los 10,000 para los *target* sólidos, con 50,000 pares para el Deuterio o *target* líquido. Esto supone una fuente de incertidumbre sustancial, pues pueden haber fluctuaciones estadísticas importantes. Es importante también considerar los efectos Coulombicos entre [0 y 0.05] GeV, que en algunos casos se hace una corrección, pero en este análisis simplemente se descarta ese tramo.¹.

Distribución Background

Si bien, la distribución *signal* es la mayor fuente de incertidumbre, la distribución *background* juega un papel importante también. Es necesario hacer una corrección de colinearidad para esta distribución como se mencionó en 2.3.3). Se observa como entre [0.1,0.4] la curva toma forma de gaussiana sesgada hacia valores más altos de T. Desde [0.4, 2] no hay diferencia visible, pero es debido a este ajuste que las funciones de correlación toman una forma descendiente.

Correcciones que podrían hacerse

Existen algunas correcciones que se pueden hacer para mejorar la precisión de los datos

- Corrección relativista: Al trabajar con velocidades cercanas a la de la luz, es necesario ajustar los ángulos medidos y algunas cantidades como el momento para tener un resultado más preciso
- *Close-Track Correction*: En experimentos de altas energías, los detectores son los encargados de medir las trayectorias de las partículas. De haber dos partículas que siguen una trayectoria muy cercana, la identificación

¹Esta es la corrección de Gamow

de la partícula se ve alterada por la calidad y resolución del detector. Para que cada partícula conserve sus características y no se “confundan” al medirse junto con otra muy cercana, es necesario hacer correcciones.

- *Acceptance Correction*: Los detectores tienen algunas limitaciones que como eficiencia de detección o cobertura angular y las correcciones de aceptación se encargan en asegurar que los detectores no limiten los resultados físicos reales.

4.2 Conclusion

Se implemento el método MEM para crear una distribución de *background* y se encontró que las funciones de correlación de los *target* presentan BEC, presentando un rango de la variable r_G de 4.308 ± 0.834 en promedio. Para el Deuterio donde hay más datos, el valor de caoticidad $\lambda = 0.6925 \pm 0.0516$. Para el resto de valores, λ llega al limite por lo que el ajuste debe ser observado con precaución o simplemente descartado.

Se observan claramente las Bose-Einstein Correlations en todos los *target*, los *ratios* son muy similares, habiendo una clara tendencia en $T < 0.2 \text{ GeV}$ de ser mayor a 2 . También notamos como T se estabiliza al valor de 1 con una leve inclinación a disminuir.

A mejorar

Para obtener resultados más completos

- Se puede implementar el método MUS para comparar con el método MEM para hacer una distribución *background*, y así contrastar estos métodos.
- Hacer todas las correcciones necesarias, como la PQ o corrección de colinealidad de la dist. *background*, *relativistic correction*, *acceptance correction*, *close-track efficiency correction*
- Se pueden presentar resultados más completos, variando entre fuentes de error como *bin*, variando entre intervalos de T, usando polinomios distintos, entre otros.
- Considerar la variable sector para hacer los cortes de vértice v_z .

Futuras investigaciones

- Hacer una nueva investigación con datos calibrados del experimento RG-E.
- Buscar y analizar BEC en dos dimensiones.

- Buscar y analizar BEC para múltiples piones por evento (3 o 4 piones), y compararlo con las correlaciones de dos piones

El objetivo principal de analizar datos de CLAS12 para buscar y encontrar correlaciones Bose-Einstein en pares de piones como primer acercamiento se llevó a cabo, donde todo parece indicar que en los *targets* se observan BEC y se induce que el radio de emisión de piones no depende del tipo de *target*.

Bibliography

- [1] S Durr et al. “Ab initio determination of light hadron masses”. In: *Science* 322.5905 (2008), pp. 1224–1227.
- [2] Jeff Greensite. *An introduction to the confinement problem*. Vol. 821. Springer, 2011.
- [3] Antonio Radic Brito. “Two-Pion Bose-Einstein Correlation measurements with CLAS detector”. In: (2019).
- [4] David J Gross and Frank Wilczek. “Asymptotically free gauge theories. I”. In: *Physical Review D* 8.10 (1973), p. 3633.
- [5] David J Gross and Frank Wilczek. “Asymptotically free gauge theories. II”. In: *Physical Review D* 9.4 (1974), p. 980.
- [6] Gerson Goldhaber et al. “Pion-pion correlations in antiproton annihilation events”. In: *Physical Review Letters* 3.4 (1959), p. 181.
- [7] Gerson Goldhaber et al. “Influence of Bose-Einstein statistics on the antiproton-proton annihilation process”. In: *Physical Review* 120.1 (1960), p. 300.
- [8] R HANBURY BROWN and RQ TWISS. “A Test of a New Type of Stellar Interferometer on Sirius”. In: *Nature* 178.4541 (1956), pp. 1046–1048.
- [9] MG Bowler. “Bose Einstein symmetrisation, coherence and chaos; with particular application to $e^+ e^-$ annihilation”. In: *Zeitschrift für Physik C Particles and Fields* 29 (1985), pp. 617–629.
- [10] LS Osborne et al. “Electroproduction of hadrons from nuclei”. In: *Physical Review Letters* 40.25 (1978), p. 1624.
- [11] Big Bubble Chamber Neutrino Collaboration et al. “Bose-Einstein correlations in neutrino and antineutrino interactions with nucleons”. In: *Zeitschrift für Physik C Particles and Fields* 60 (1993), pp. 37–51.
- [12] S Chekanov et al. “Bose–Einstein correlations in one and two dimensions in deep inelastic scattering”. In: *Physics Letters B* 583.3-4 (2004), pp. 231–246.
- [13] A Airapetian et al. “Bose–Einstein correlations in hadron-pairs from lepto-production on nuclei ranging from hydrogen to xenon: HERMES Collaboration”. In: *The European Physical Journal C* 75 (2015), pp. 1–9.

- [14] ALICE Collaboration et al. “Multipion Bose-Einstein correlations in pp, p-Pb, and Pb-Pb collisions at the LHC”. In: *arXiv preprint arXiv:1512.08902* (2015).
- [15] Ghulam Bary et al. “Analyses of multi-pion Bose-Einstein correlations for granular sources with coherent pion-emission droplets”. In: *Chinese Physics C* 45.2 (2021), p. 024106.
- [16] Lesya Shchutska. “Study of the Bose-Einstein correlations of same-sign pions in proton-lead collisions”. In: (2023).
- [17] Bernhard A Mecking et al. “The CEBAF large acceptance spectrometer (CLAS)”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 503.3 (2003), pp. 513–553.
- [18] VD Burkert et al. “The CLAS12 spectrometer at Jefferson laboratory”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 959 (2020), p. 163419.
- [19] Hayk Hakobyan et al. “A double-target system for precision measurements of nuclear medium effects”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 592.3 (2008), pp. 218–223.
- [20] ROOT Team. *ROOT: analyzing petabytes of data, scientifically*. <https://root.cern/>. Accessed: 01 january 2025.

A

Anexo A — Códigos C++

A.1 FilterNtupleDIS.C

Código encargado de filtrar los archivos root, usando las variables y criterios mencionados en la sección 2.3.1)

```
#include <TFile.h>
#include <TTree.h>
#include <TString.h>
#include <iostream>

// CUTS para un archivo root.
//-----
// DIS: Q2 > 1 && W2>4 && y_bjorken < 0.85
// TARGET: -9<v_z<-3.5 ;D2 || -2.5<v_z<3; C,Al,Cu,Sn,Pb
// PARTICLE: pid==211; pion || pid==11; electron
// INPUT: Nombre archivo root, Nombre target
// OUTPUT: Archivo filtrado
void FilterNtupleDIS(const char* inputFileName, const char* elementSymbol) {
//EXECUTE
// .x macrotolina/FilterNtupleDIS.C("ntuples_dc_020082.root", "D2")
    const char* inputFolder = "RG_E";    // input folder
    const char* outputFolder = "FiltradosDIS"; // output folder

    // input path
    TString inputFilePath = TString::Format("%s/%s", inputFolder, inputFileName);

    // input file
    TFile* inputFile = TFile::Open(inputFilePath, "READ");
```

```

// Get the ntuple
TTree* tree = dynamic_cast<TTree*>(inputFile->Get("data"));

// CUT
std::string cut;
if (std::string(elementSymbol) == "D2") {
    cut = "Q2 > 1 & W2>4 & y_bjorken < 0.85 & v_z < -3.5 & v_z > -9 & pid == 211";
} else {
    cut = "Q2 > 1 & W2>4 & y_bjorken < 0.85 & v_z < 3 & v_z > -2.5 & pid == 211";
}

// output Nombre de archivo
TString outputFileName = inputFileName;
TString symbol(elementSymbol);
outputFileName.ReplaceAll("ntuples_dc_", "clas12_");
outputFileName.ReplaceAll("_", "_" + symbol + "_");

// Output Path
TString outputFilePath = TString::Format("%s/%s", outputFolder, outputFileName.Data());

// output file
TFile* outputFile = TFile::Open(outputFilePath, "RECREATE");

// Cut, cout
std::cout << "Filtrando:" << cut << std::endl;

// New Tree using cut
TTree* filteredTree = tree->CopyTree(cut.c_str());

// write file
outputFile->cd();
filteredTree->Write("ntuple");

// Clean
inputFile->Close();
outputFile->Close();

std::cout << "Archivo creado: " << outputFilePath << std::endl;
}

```

A.2 MergeNtuples.C

Código encargado en fusionar las distintas *runs* que compartan tipo de *target*, de la sección 2.3.2).

```
#include <TChain.h>
#include <TSystem.h>
#include <TFile.h>
#include <iostream>
#include <vector>
#include <string>

// MERGE de archivos ROOT con TTree
// -----
// INPUT: Carpeta con archivos ROOT filtrados
// OUTPUT: Archivo ROOT combinado con todos los ntuples
void MergeNtuples(const char* inputFolder, const char*
outputFolder, const char* element) {
    // EXECUTE
    // .x macrotesina/MergeNtuples.C("D2_DISfiltered","Merged","D2")

    // Lista de archivos ROOT en el inputFolder con el elemento dado
    void* dir = gSystem->OpenDirectory(inputFolder);
    const char* fileName;
    std::vector<std::string> fileList;

    while ((fileName = gSystem->GetDirEntry(dir))) {
        std::string fullFileName = std::string(inputFolder) + "/" + fileName;
        if (fullFileName.find(element) !=
            std::string::npos && fullFileName.find(".root") != std::string::npos) {
            fileList.push_back(fullFileName);
        }
    }
    gSystem->FreeDirectory(dir);

    // Verificación de archivos
    if (fileList.empty()) {
        std::cerr << "Error: No hay archivos con " <<
            element << " encontrados en " << inputFolder << std::endl;
        return;
    }

    // Creación de TChain
    TChain chain("ntuple");
```

```

// Agregar archivos al TChain
for (const auto& file : fileList) {
    chain.Add(file.c_str());
    std::cout << "Se agregó archivo: " << file << std::endl;
}

// Número total de entries
Long64_t nentries = chain.GetEntries();
std::cout << "# total de entries: " << nentries << std::endl;

// Nombre del archivo de salida
std::string outputFileName =
std::string(outputFolder) + "/merged_" + element + ".root";

// Fusión de los archivos ROOT
if (chain.Merge(outputFileName.c_str()) > 0) {
    std::cout << "Merged file guardado como '" << outputFileName << "'" << std::endl;
} else {
    std::cerr << "Error: Merging fallido!" << std::endl;
}
}

```

A.3 FilterPairsMerged.C

Código usado para obtener eventos con un par de piones positivos, mencionado en la sección 2.3.2

```

#include <TFile.h>
#include <TTree.h>
#include <TString.h>
#include <iostream>

// FILTRO de eventos con exactamente dos partículas en archivos ROOT fusionados
// -----
// INPUT: Archivo ROOT fusionado con eventos
// OUTPUT: Archivo ROOT con eventos que contienen exactamente dos partículas
void FilterPairsMerged(const char* inputFileName) {
    // EXECUTE
    // .x macrotesina/FilterPairsMerged.C("merged_D2.root")

    const char* inputFolder = "merged"; // Carpeta de entrada
    const char* outputFolder = "mergedpairs"; // Carpeta de salida

    // Construcción de paths

```

```

TString inputFilePath = TString::Format("%s/%s", inputFolder, inputFileName);

// Archivo de entrada y TTree
TFile* inputFile = TFile::Open(inputFilePath, "READ");
TTree* tree = dynamic_cast<TTree*>(inputFile->Get("ntuple"));

// Variable para almacenar el número de evento
Float_t event_num;
tree->SetBranchAddress("event_num", &event_num);

// Archivo de salida
TString outputFileName = TString(inputFileName).ReplaceAll("merged", "mpair2");
TString outputFilePath = TString::Format("%s/%s", outputFolder, outputFileName.Data());
TFile* outputFile = TFile::Open(outputFilePath, "RECREATE");

// Creación de un nuevo TTree con eventos filtrados
TTree* filteredTree = tree->CloneTree(0);
Long64_t nEntries = tree->GetEntries();
Long64_t i = 0;

// ALGORITMO: FILTRO DE EVENTOS CON EXACTAMENTE DOS PARTÍCULAS
while (i < nEntries) {
    tree->GetEntry(i);
    Float_t currentEvent = event_num;
    Int_t count = 0;
    Long64_t start = i;

    // Conteo de partículas en un evento
    while (i < nEntries) {
        tree->GetEntry(i);
        if (event_num != currentEvent) break; // Cambio de evento
        ++count;
        ++i;
    }

    // Se guarda solo si hay exactamente dos partículas en el evento
    if (count == 2) {
        for (Long64_t j = start; j < i; ++j) {
            tree->GetEntry(j);
            filteredTree->Fill();
        }
    }
    i++;
}

// Guardado del archivo de salida

```

```

    outputFile->cd();
    filteredTree->Write("ntuple");
    inputFile->Close();
    outputFile->Close();

    std::cout << "Archivo de pares creado: " << outputPath << std::endl;
}

```

A.4 expT.C

Código usado para calcular la variable T para un par de piones correlacionados, para formar la distribución *signal*. Detalles en la sección 2.3.3)

```

#include <TFile.h>
#include <TTree.h>
#include <TLorentzVector.h>
#include <TVector3.h>
#include <TMath.h>
#include <iostream>
#include <string>

// Esta macro calcula T en coordenadas esféricas
// para alinear la segunda partícula respecto a la primera.
// INPUT: Archivo ROOT con datos de piones emparejados.
// OUTPUT: Archivo ROOT con el cálculo de T.
void expT(const char* target) {
    // EXECUTE
    // .x macrotresina/expT.C("D2")

    // Constantes
    const Float_t pi_mass = 0.13957018; // Masa del pión en GeV/c^2

    // Rutas de archivos
    std::string inputFolder = "mergedpairs/";
    std::string outputFolder = "final/";
    std::string inputFileName = inputFolder + "mpair2_" + target + ".root";
    std::string outputFileName = outputFolder + "expT_" + target + ".root";

    // Archivo de entrada
    TFile *inputFile = TFile::Open(inputFileName.c_str(), "READ");
    TTree *inputTree = (TTree*)inputFile->Get("ntuple");

    // Variables de entrada
    Float_t p, theta, phi, event_num, Q2, nu;

```

```

inputTree->SetBranchAddress("p", &p);
inputTree->SetBranchAddress("theta", &theta);
inputTree->SetBranchAddress("phi", &phi);
inputTree->SetBranchAddress("event_num", &event_num);

// Archivo de salida
TFile *outputFile = TFile::Open(outputFileName.c_str(), "RECREATE");
TTree *outputTree = new TTree("ntuple", "Tree con T calculado");

// Variables de salida
Float_t T, P1, PX1, PY1, PZ1, P2, PX2, PY2, PZ2;
Float_t Energy1, Energy2, q4;
Float_t theta_PQ1, phi_PQ1;

outputTree->Branch("T", &T, "T/F");
outputTree->Branch("P1", &P1, "p1/F");
outputTree->Branch("PX1", &PX1, "px1/F");
outputTree->Branch("PY1", &PY1, "py1/F");
outputTree->Branch("PZ1", &PZ1, "pz1/F");
outputTree->Branch("P2", &P2, "p2/F");
outputTree->Branch("PX2", &PX2, "px2/F");
outputTree->Branch("PY2", &PY2, "py2/F");
outputTree->Branch("PZ2", &PZ2, "pz2/F");

// ALGORITMO: Cálculo de T para cada par con transformación de ángulos PQ
Float_t current_event = -1;
bool first_particle = true;
TVector3 p1, p2;

Long64_t nentries = inputTree->GetEntries();
for (Long64_t i = 0; i < nentries; ++i) {
    inputTree->GetEntry(i);

    if (event_num != current_event) {
        current_event = event_num;
        first_particle = true;
    }

    if (first_particle) {
        Energy1 = TMath::Sqrt(TMath::Power(p, 2) + TMath::Power(pi_mass, 2));
        p1.SetMagThetaPhi(p, theta, phi);

        first_particle = false;
    } else {

```

```

    Energy2 = TMath::Sqrt(TMath::Power(p, 2) + TMath::Power(pi_mass, 2));
    p2.SetMagThetaPhi(p, theta, phi);

    P1 = p1.Mag();
    PX1 = p1.X();
    PY1 = p1.Y();
    PZ1 = p1.Z();

    P2 = p2.Mag();
    PX2 = p2.X();
    PY2 = p2.Y();
    PZ2 = p2.Z();

    q4 = TMath::Power(Energy1 - Energy2, 2) -
        ((TMath::Power(PX1 - PX2, 2)) +
         (TMath::Power(PY1 - PY2, 2)) +
         (TMath::Power(PZ1 - PZ2, 2)));
    T = TMath::Sqrt(-q4);

    outputTree->Fill();
}
}

outputTree->Write();
outputFile->Close();
inputFile->Close();

std::cout << "Se calculó T y se guardó como " << outputFileName << std::endl;
}

```

A.5 MEMspherical.C

Código usado para calcular la variable T entre piones no correlacionados para formar la distribución *background*). Explicado en la sección 2.3.3

```

#include <iostream>
#include <vector>
#include <TRandom3.h>
#include <TChain.h>
#include <TNtuple.h>
#include <TMath.h>
#include <TVector3.h>
#include <TSystem.h>

```

```

// Calculo de distribucion background con metodo MEM. No Usa correccion pq
// -----
// INPUT: Tipo de Target, y cantidad de pares no correlacionados
// OUTPUT: Archivo con el numero de pares no correlacionados deseado.
void MEMspherical(TString target, Int_t max_pairs = 1e6) {
    // EXECUTE
    // .x macrotesina/MEMspherical.C("D2", 1e6)

    const Float_t pi_mass = 0.13957018; // Masa del pion en GeV/c^2

    // Paths
    TString input_folder = "mergedpairs";
    TString output_folder = "Background";
    TString input_file = Form("%s/mpair2_%s.root", input_folder.Data(), target.Data());
    TString output_file = Form("%s/MEMsph_%s.root", output_folder.Data(), target.Data());

    // Input TChain
    TChain files("ntuple");
    files.Add(input_file);

    // Variables de ntuple
    Float_t p, theta, phi, event_num, run_num;
    files.SetBranchAddress("p", &p);
    files.SetBranchAddress("theta", &theta);
    files.SetBranchAddress("phi", &phi);
    files.SetBranchAddress("event_num", &event_num);
    files.SetBranchAddress("run_num", &run_num);

    // Output TNtuple
    const char *VarList = "T:P1:PX1:PY1:PZ1:P2:PX2:PY2:PZ2";
    TNtuple background_tuple("mixed_pairs", "Mixed Event Pairs", VarList);

    // RNG
    TRandom3 rand_gen;
    rand_gen.SetSeed();

    // Variables de los pares
    Float_t T, P1, PX1, PY1, PZ1, P2, PX2, PY2, PZ2;
    Int_t pair_count = 0;

    Int_t nentries = files.GetEntries();
    std::cout << "Total entries in input file: " << nentries << std::endl;

    // Almacenar eventos en un vector
    std::vector<std::vector<Float_t>> events;

```

```

for (Int_t i = 0; i < nentries; i++) {
    files.GetEntry(i);
    events.push_back({p, theta, phi, event_num, run_num});
}

while (pair_count < max_pairs) {
    // Seleccionar pares aleatorios
    Int_t event1_index = rand_gen.Integer(nentries);
    Int_t event2_index = rand_gen.Integer(nentries);

    // Excluir pares inválidos
    if (event1_index == event2_index) continue;

    auto &event1 = events[event1_index];
    auto &event2 = events[event2_index];

    if (event1[3] == event2[3] && event1[4] == event2[4]) continue;

    // Crear el vector de momento para la primera partícula
    TVector3 p1;
    p1.SetMagThetaPhi(event1[0], event1[1], event1[2]);

    // Crear el vector de momento para la segunda partícula
    TVector3 p2;
    p2.SetMagThetaPhi(event2[0], event2[1], event2[2]);

    // Extraer componentes de los vectores de momento ya rotados
    P1 = p1.Mag();
    PX1 = p1.X();
    PY1 = p1.Y();
    PZ1 = p1.Z();

    P2 = p2.Mag();
    PX2 = p2.X();
    PY2 = p2.Y();
    PZ2 = p2.Z();

    // Calculo T (Diferencia de cuadrimomento)
    Float_t Energy1 = TMath::Sqrt(TMath::Power(P1, 2) + TMath::Power(pi_mass, 2));
    Float_t Energy2 = TMath::Sqrt(TMath::Power(P2, 2) + TMath::Power(pi_mass, 2));
    Float_t q4 = TMath::Power(Energy1 - Energy2, 2) -
        (TMath::Power(PX1 - PX2, 2) +
         TMath::Power(PY1 - PY2, 2) +

```

```

        TMath::Power(PZ1 - PZ2, 2));
T = TMath::Sqrt(-q4);

// Llenar el ntuple
background_tuple.Fill(T, P1, PX1, PY1, PZ1, P2, PX2, PY2, PZ2);
pair_count++;
}

// Archivo de salida
TFile outfile(output_file, "RECREATE");
background_tuple.Write();
outfile.Close();

std::cout << "Background corrected file created: " << output_file << std::endl;
std::cout << "Total pairs generated: " << pair_count << std::endl;
}

```

A.6 hratioBEC3.C

Código usado para calcular el *ratio* entre las distribuciones *signal*, y *background*, y luego haciendo un ajuste con la ecuación 1.12)

```

#include <TNtuple.h>
#include <TFile.h>
#include <TH1F.h>
#include <TCanvas.h>
#include <TF1.h>
#include <TStyle.h>
#include <TSystem.h>
#include <ctime>

// Calcula la función de correlación Bose-Einstein a partir de una
// distribución signal y background y guarda solo la imagen generada.
void hratioBEC3(const char* target, int bins = 80,
float lbound = 0.05, float hbound = 1.6) {
    // EXECUTE
    // .x macrotesina/hratioBEC3.C("D2", 80, 0, 1.6)

    // Obtener la fecha actual (día y mes)
    time_t now = time(0);
    struct tm *ltm = localtime(&now);
    char timestamp[16];
    snprintf(timestamp, sizeof(timestamp), "%02d%02d",
ltm->tm_mday, ltm->tm_mon + 1);

```

```

// Definir histogramas
TH1F *expD = new TH1F("D_s", "Signal Distribution;
T; Events", bins, lbound, hbound);
TH1F *backD = new TH1F("D_b", "Background
Distribution; T; Events", bins, lbound, hbound);
TH1F *hist_ratio = new TH1F("R(T)", Form("BEC Ratio
for %s; T; R(T)", target), bins, lbound, hbound);

expD->Sumw2();
backD->Sumw2();
hist_ratio->Sumw2();

// Rutas de archivos
std::string expFileName = "final/expT_" +
std::string(target) + ".root";
std::string backFileName = "background/MEMsph_" +
std::string(target) + ".root";
std::string outputImageName =
"GraficosTESINA/BECratio_" + std::string(target) +
"_" + timestamp + ".png";

// Abrir archivos de entrada
TFile *expFile = TFile::Open(expFileName.c_str(),
"READ");
if (!expFile || expFile->IsZombie()) {
    std::cerr << "Error: No se pudo abrir el
    archivo signal " << expFileName << std::endl;
    return;
}
TTree *expTree = (TTree*)expFile->Get("ntuple");

TFile *backFile = TFile::Open(backFileName.c_str(),
"READ");
if (!backFile || backFile->IsZombie()) {
    std::cerr << "Error: No se pudo abrir el
    archivo background " << backFileName <<
    std::endl;
    expFile->Close();
    return;
}
TTree *backTree = (TTree*)backFile-
>Get("mixed_pairs");

// Llenar histogramas
Float_t T_e, T_b;

```

```

expTree->SetBranchAddresses("T", &T_e);
backTree->SetBranchAddresses("T", &T_b);

Long64_t nEntries_exp = expTree->GetEntries();
for (Long64_t i = 0; i < nEntries_exp; i++) {
    expTree->GetEntry(i);
    expD->Fill(T_e);
}

Long64_t nEntries_back = backTree->GetEntries();
for (Long64_t i = 0; i < nEntries_back; i++) {
    backTree->GetEntry(i);
    backD->Fill(T_b);
}

// Normalización
if (expD->Integral() > 0) expD->Scale(1.0 / expD-
>Integral());
if (backD->Integral() > 0) backD->Scale(1.0 / backD-
>Integral());

// Calcular el ratio
hist_ratio->Divide(expD, backD, 1.0, 1.0, "B");

// Excluir T^2 en un intervalo específico
for (int bin = 1; bin <= hist_ratio->GetNbinsX();
bin++) {
    double x = hist_ratio->GetBinCenter(bin);
    if (x >= 0.0 && x <= 0.125) {
        hist_ratio->SetBinContent(bin, 0);
        hist_ratio->SetBinError(bin, 0);
    }
}

// Definir función de ajuste
TF1 *f1 = new TF1("f1", "[0] * (1 + [1] * exp(-[2]*
[2] * x*x)) * (1 + [3] * x + [4] * x*x)", 0.125,
hbound);
f1->SetParNames("gamma", "lambda", "r", "delta",
"epsilon");
f1->SetParameters(1.142, 0.653, 0.533, -0.393,
0.263);
f1->SetParLimits(1, 0.01, 1.0); // lambda
f1->SetParLimits(2, 0.5, 10.0); // r
TVirtualFitter::Fitter(hist_ratio)-

```

```

>SetMaxIterations(10000);
f1->SetNpx(500);
hist_ratio->Fit("f1", "RMS");
gStyle->SetOptFit(1111);

// Estilos del histograma
double minY = hist_ratio->GetBinContent(hist_ratio-
>GetMinimumBin()); // Mínimo real
double maxY = hist_ratio->GetMaximum() * 1.1; //
Máximo con un 10% extra

hist_ratio->GetYaxis()->SetRangeUser(0.5, maxY);
hist_ratio->SetMarkerStyle(20);
hist_ratio->SetMarkerSize(1.2);
hist_ratio->SetLineWidth(2);
hist_ratio->SetLineColor(kBlack);

// Dibujar y guardar imagen
TCanvas *c1 = new TCanvas("c1", "BEC Ratio", 1900,
1200);
hist_ratio->Draw("E2");
hist_ratio->Draw("E1 SAME");
c1->Modified();
c1->Update();
f1->Draw("same");
c1->SaveAs(outputImageName.c_str());

// Liberar memoria
delete expD;
delete backD;
delete hist_ratio;
delete c1;
delete f1;
expFile->Close();
backFile->Close();
delete expFile;
delete backFile;
}

```