

Universidade de São Paulo
Instituto de Física

Estudo da reação de breakup ${}^4\text{He}({}^{17}\text{F}, {}^{16}\text{O}+p){}^4\text{He}$
usando o alvo ativo pAT-TPC: uma abordagem
usando técnicas de Machine Learning

Guilherme Ferrari Fortino

Orientador: Prof. Dr. Valdir Guimarães _____

Coorientador: Dr. Juan Carlos Zamora Cardona _____

Dissertação de mestrado apresentada ao Instituto de
Física da Universidade de São Paulo, como requisito
parcial para a obtenção do título de Mestre(a) em Ciências.

Banca Examinadora:

Prof(a). Dr(a). Nome do(a) Professor(a) - Orientador (instituição de trabalho)

Prof(a). Dr(a). Nome do(a) Professor(a) (instituição de trabalho)

Prof(a). Dr(a). Nome do(a) Professor(a) (instituição de trabalho)

São Paulo
2022

Resumo

Neste trabalho estudamos a reação de breakup ${}^4\text{He}({}^{17}\text{F}, {}^{16}\text{O} + p){}^4\text{He}$ usando o alvo ativo pAT-TPC. A análise dos dados experimentais foi realizada usando técnicas de machine learning que permitiram analisar de forma eficiente um grande volume de dados. Os algoritmos desenvolvidos neste trabalho correspondem à primeira aplicação prática de técnicas de machine learning numa análise de dados com alvo ativo.

Na primeira parte da análise foi feita a reconstrução tridimensional dos eventos através da análise dos pulsos gerados pelo plano detector micromegas. O micromegas é um detector multipixelado (2048 canais), onde cada pixel (com coordenadas x e y fixas) é um canal do detector com eletrônica independente. Foram criadas três redes neurais supervisionadas para analisar os pulsos “crus” envolvendo as seguintes etapas: identificação do fundo, deconvolução e identificação dos centróides e carga dos pulsos. Esta informação foi de grande importância para reconstruir milhões de nuvens de pontos que correspondem aos eventos das reações nucleares no alvo ativo.

Já com as nuvens de pontos reconstruídas, as trajetórias das partículas foram identificadas usando algoritmos de clustering e estimadores robustos. A partir das propriedades geométricas das trajetórias, foi calculado o vértice de reação de cada evento para enfim poder obter os ângulos de espalhamento da reação. A identificação dos prótons foi feita a partir do comprimento e energia depositada por cada trajetória. Identificado o canal de breakup, as distribuições angulares (breakup inclusivo e exclusivo) foram construídas. Por fim, as distribuições angulares foram analisadas e comparadas com resultados da literatura envolvendo o breakup do ${}^{17}\text{F}$ em outros núcleos alvo.

Abstract

In this work, the breakup reaction ${}^4\text{He}({}^{17}\text{F}, {}^{16}\text{O} + p){}^4\text{He}$ was investigated using the active target pAT-TPC. The data analysis was performed with machine learning techniques, which allowed to process a large amount of data in a very efficient way. The algorithms developed in this work represent the first practical application of machine learning techniques for the analysis of an experiment using an active target.

In the first part of the analysis, a three-dimensional reconstruction of the events was performed through the analysis of the pulses generated by the micromegas pad plane. The micromegas pad plane is a highly-segmented detector (2048 pixels), where each individual pad (with certain x and y coordinates) has an independent electronics. Three supervised neural networks were created to perform the analysis of the raw data, including the following parts: background identification, deconvolution, and centroid and charge identification. This information was of great importance in order to reconstruct millions of point clouds which correspond to the detected nuclear reactions in the active target.

The reconstructed point clouds were analyzed with clustering algorithms and robust estimators in order to identify the individual particle trajectories. The respective reaction vertices and scattering angles were obtained by using geometric properties of the reconstructed trajectories. The identification of the proton tracks was performed by using the particle range and deposited energy for each track. Angular distributions of inclusive and exclusive breakup were extracted from the analysis of the point clouds with only proton tracks. Finally, the angular distributions were analyzed and compared with data from the literature involving the breakup of ${}^{17}\text{F}$ on heavier target nuclei.

Sumário

1	Introdução	6
2	O experimento	9
2.1	Produção do feixe secundário ^{17}F usando o sistema TWINSOL	9
2.2	O alvo ativo pAT-TPC	13
3	Desenvolvimento de ferramentas de <i>machine learning</i> para análise de dados	20
3.1	Tipos de redes neurais	21
3.2	Estrutura da rede neural	25
3.3	Sistemas de <i>machine learning</i>	27
4	Reconstrução de nuvens de pontos a partir de algoritmos de <i>machine learning</i>	29
4.1	Construção do banco de dados para as redes neurais	30
4.1.1	Estimativa do fundo	32
4.1.2	Deconvolução do sinal	34
4.2	Análise dos pulsos com <i>machine learning</i>	36
4.2.1	Rede neural para o fundo	36
4.2.2	Rede neural para a deconvolução	40
4.2.3	Identificação de picos	43
4.2.4	Acoplando as redes neurais	46
5	Análise das nuvens de pontos	49
5.1	Identificação de trajetórias	49
5.2	Abordagens alternativas	55
5.2.1	Identificação de eventos com <i>machine learning</i>	55
5.2.2	Identificação de <i>outliers</i>	58
6	Resultados	60
6.1	A cinemática da reação	60

6.1.1	Reconstrução do vértice da reação	60
6.1.2	Reconstrução da energia das partículas	62
6.1.3	Determinação dos ângulos de reação	64
6.2	Identificação do canal de reação de breakup	64
6.3	Construção das distribuições angulares	71
6.4	Análise e comparação dos resultados	77
7	Conclusões	82
A	Aplicações de machine learning na física nuclear	84
A.1	Análise de espectros para identificação de partículas (<i>particle identification</i> , PID)	84
A.2	Estimativa de raios e massas nucleares	86
A.3	Decaimento β e processo r	87
A.4	Alvos ativos	88
B	Algoritmo p-RANSAC	91

Capítulo 1

Introdução

Um dos principais objetivos das investigações realizadas na área de física nuclear é entender a estrutura do núcleo. Apesar do sucesso de vários modelos, tais como modelo de camadas, para explicar as estruturas de núcleos estáveis, os núcleos instáveis ou exóticos, ricos ou pobres em nêutrons, continuam sendo um grande desafio para nossa compreensão.

Os núcleos leves radioativos são de grande interesse para a astrofísica nuclear [1], onde processos de captura rápida de nêutrons e prótons (que ocorrem em eventos explosivos de novas ou supernovas) envolvem os núcleos radioativos longe da região de estabilidade. Conhecer as propriedades desses núcleos, como a massa, probabilidades de decaimento, seções de choque de captura e de breakup é fundamental para o entendimento da vários eventos astrofísicos [1, 2]. A figura 1.1 mostra parte da tabela de nuclídeos, onde é mostrado núcleos longe da região de estabilidade, com a informação da energia de separação do próton (S_p).

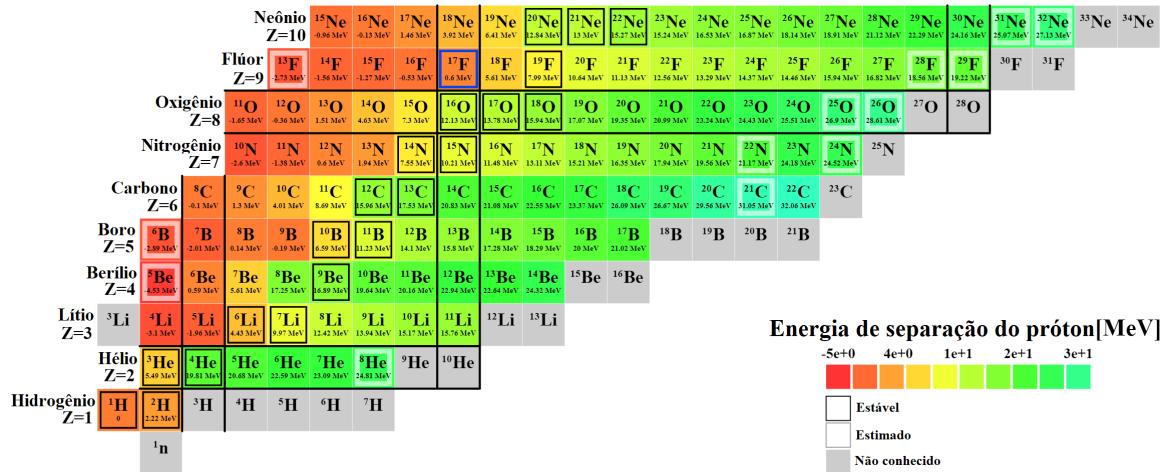


Figura 1.1: Tabela de nuclídeos para elementos leves, indicando a energia de separação de um próton em MeV. O núcleo estudado nesse trabalho é o ^{17}F (destacado pelo quadrado em cor azul), com $S_p = 600$ keV. [3]

Para investigar a estrutura de núcleos ricos em prótons ou em nêutrons, pode-se usar reações nucleares como o espalhamento elástico, transferência ou breakup [4]. Em particular, a reação de breakup é de grande importância para o estudo da estrutura dos núcleos radioativos ricos em prótons ou em nêutrons que possuem uma forte estrutura de cluster do tipo caroço + partículas de valência, onde as partículas de valência são fracamente ligadas [4, 5]. Quando a energia de ligação dessas partículas de valência é muito baixa, da ordem de centenas de keV até poucos MeVs, elas podem orbitar o caroço a grandes distâncias, formando uma configuração conhecida como "estrutura halo". A reação estudada nessa dissertação é a reação de breakup $^4\text{He}(^{17}\text{F}, ^{16}\text{O}+p)^4\text{He}$, onde o ^{17}F é um núcleo rico em prótons que forma uma estrutura halo formada por um núcleo que pode ser descrito como um núcleo de ^{16}O mais um próton fracalemente ligado ($S_p = 600$ keV) [5].

Para melhor caracterizar o mecanismo de uma reação de breakup, a identificação e detecção de ambos fragmentos da dissociação é muito importante. Para tanto equipamentos e sistemas de detecção complexos e sofisticados tem sido desenvolvidos. Um desses equipamentos é o Alvo Ativo. Conforme está explicado com detalhes ao longo dessa dissertação, o alvo ativo é um aparato que funciona tanto como alvo como sistema de detecção. O alvo ativo permite o uso de um gás específico que serve tanto quanto alvo quanto meio detector. Isso permite a detecção de múltiplas reações nucleares ao mesmo tempo, com a capacidade de medidas completas da cinemática das reações [6, 7, 8]. Com o alvo ativo é possível determinar as trajetórias tridimensionais das partículas envolvidas numa reação nuclear e reconstruir o respectivo vértice (posição da reação).

Nos últimos anos, alvos ativos envolvendo câmaras de projeção no tempo, *active-target time projection chambers* (AT-TPCs), se tornaram importantes e relevantes para o estudo

de núcleos exóticos em física nuclear [6]. Diversos desses tipos de equipamentos foram desenvolvidos em diferentes laboratórios de física nuclear pelo mundo [8, 9, 10].

O sistema de detecção de TPCs, usualmente, é baseado na ideia de se detectar os elétrons provenientes da ionização do gás. Para tanto, foi desenvolvido um dispositivo que combina o micromegas (Micro-MEsh Gaseous Structures) [11] e o GEM (Gas Electron Multiplier) [12]. Esse dispositivo possui uma alta granularidade do plano detector, da ordem de 10^3 a 10^4 canais [6]. Com isso, experimentos que envolvem TPCs produzem enormes quantidades de dados [8, 13]. Por exemplo, o experimento analisado neste trabalho gerou cerca de 3 TB de dados crus em alguns dias de medida. Devido à grande quantidade de dados produzidos, a análise precisa ser dividida em diversas etapas, de modo que a eficiência e tempo computacional têm papel fundamental no processo. Nesse sentido, o uso de algoritmos como de machine learning pode trazer vantagens significativas para a análise. Esses algoritmos, por serem muito mais rápidos do que análise baseadas em algoritmos de programação convencional, podem viabilizar a análise dessa quantidade de dados num tempo razoável. Um dos objetivos desse trabalho foi exatamente o de desenvolver esse tipo de técnica para análise de dados obtidos de experimentos com alvos ativos. Alguns desses códigos e parte da análise desta dissertação já foram publicados em revistas indexadas internacionais [6, 14].

A utilização de técnicas de machine learning já está sendo amplamente utilizada nas mais diversas áreas de nossa vida cotidiana (softwares de detecção de spam, sistemas de recomendação, marcação em fotos de redes sociais, assistentes pessoais ativados por voz, carros autônomos, smartphones com reconhecimento facial e muito mais). Essas técnicas estão agora sendo desenvolvidas e aplicadas para análise de dados obtidos de experiências científicas.

A dissertação está esquematizada da seguinte forma: no capítulo 2 está feita a descrição do experimento cujos dados foram analisados; no capítulo 3 está feita uma breve descrição do que é machine learning com exemplos de algoritmos e métodos utilizados; O capítulo 4 foi dedicado para a análise dos sinais (pulsos) medidos no experimento; No capítulo 5 está descrita a análise das nuvens de pontos reconstruídas a partir dos pulsos; No capítulo 6 foi feita a construção das distribuições angulares bem como a análise e comparação com experimentos da literatura relacionados; Por fim, no capítulo 7 está a conclusão do trabalho.

Capítulo 2

O experimento

Esse trabalho corresponde a análise de dados obtidos da experiência onde foi medida a reação de breakup para o sistema $^{17}\text{F} + ^4\text{He}$. Nessa experiência utilizamos um feixe de ^{17}F , que é um núcleo radioativo com tempo de vida média de 64.37 segundos, incidindo sobre um alvo gasoso de ^4He . A experiência foi realizada em outubro de 2019 na University of Notre Dame, Estados Unidos. O feixe radioativo de ^{17}F foi obtido com o sistema de produção chamado TWINSOL [15] e conduzido até o alvo ativo pAT-TPC (prototype Active Target - Time Projection Chamber) [13]. Neste capítulo está descrito como o feixe radioativo ^{17}F foi obtido e como os produtos da reação de breakup foram medidos.

2.1 Produção do feixe secundário ^{17}F usando o sistema TWINSOL

O feixe radioativo ^{17}F foi produzido em voo usando o sistema TWINSOL [16, 17]. Um feixe primário de ^{16}O foi acelerado a uma energia de 70 MeV e conduzido até a câmara de produção com uma intensidade da ordem de 200 nA. O alvo de produção consiste de uma célula gasosa, preenchida com gás de deutério à uma pressão de 1 atm. A partir da reação de transferência de um próton ou um nêutron do feixe primário (^{16}O) com o alvo de produção (deutério), partículas como ^{17}F e ^{17}O foram produzidas [16, 15]. Essas partículas são selecionadas pelo sistema de duplo solenoides formando um feixe com essas partículas [15, 18]. A figura 2.1 mostra o desenho do sistema TWINSOL acoplado com o pAT-TPC.

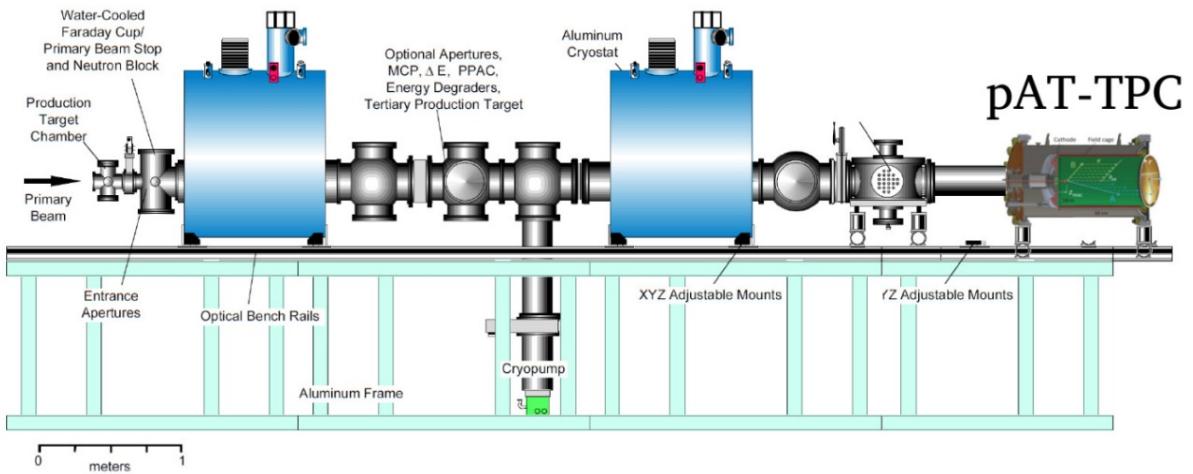


Figura 2.1: Desenho do TWINSOL à esquerda e do pAT-TPC à direita. O feixe estável de ^{16}O entra à esquerda do TWINSOL, produzindo no alvo de produção o feixe secundário ^{17}F que irá ser conduzido até o alvo ativo pAT-TPC. Todo o sistema está localizado na University of Notre Dame.

O TWINSOL é um sistema de produção de feixes radioativos em voo que possui dois solenoides supercondutores alinhados que são usados para produzir, coletar, transportar, focar e selecionar feixes estáveis e radioativos. O sistema se baseia na seleção de partículas a partir da sua rigidez magnética ($B\rho$) [15, 18, 19]. Cada solenoide possui 15 cm de raio interno e 1 m de comprimento [15]. O fato de ser um solenoide finito faz com que surjam efeitos de borda na componente radial do campo magnético do solenoide, cujo efeito é fazer com que o solenoide seja capaz de selecionar e focalizar partículas [18]. Para entender melhor o efeito de borda no campo magnético, e consequentemente o funcionamento do TWINSOL, simulações computacionais usando a biblioteca GEANT4 [20] foram feitas usando a geometria do sistema “irmão” do TWINSOL, o *Radioactive Ion Beams in Brasil* (RIBRAS), que também possui dois solenoides supercondutores alinhados [18, 21]. A figura 2.2 mostra a geometria usada na simulação, onde os solenoides são de cor verde focalizando as partículas de cor laranja em um ponto do plano focal em vermelho. O campo magnético usado na simulação é função da posição do eixo, de cada solenoide, e está mostrado na figura 2.3.

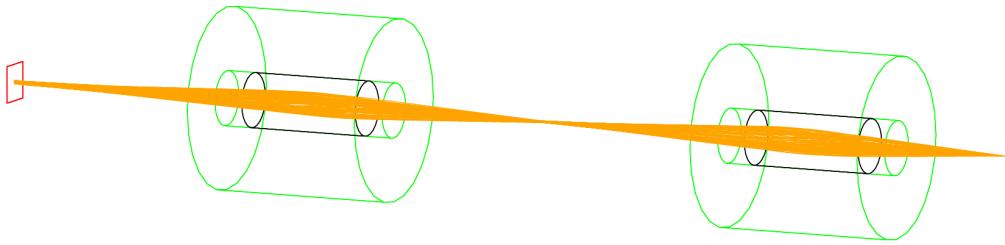


Figura 2.2: Simulação computacional do sistema RIBRAS, onde as partículas carregadas em laranja surgem do ponto à direita da figura e passam pelos dois solenoides em verde, para então serem focalizada em um ponto do plano focal em vermelho. A parte em preto dos solenoides corresponde aos limites físicos da bobina [18].

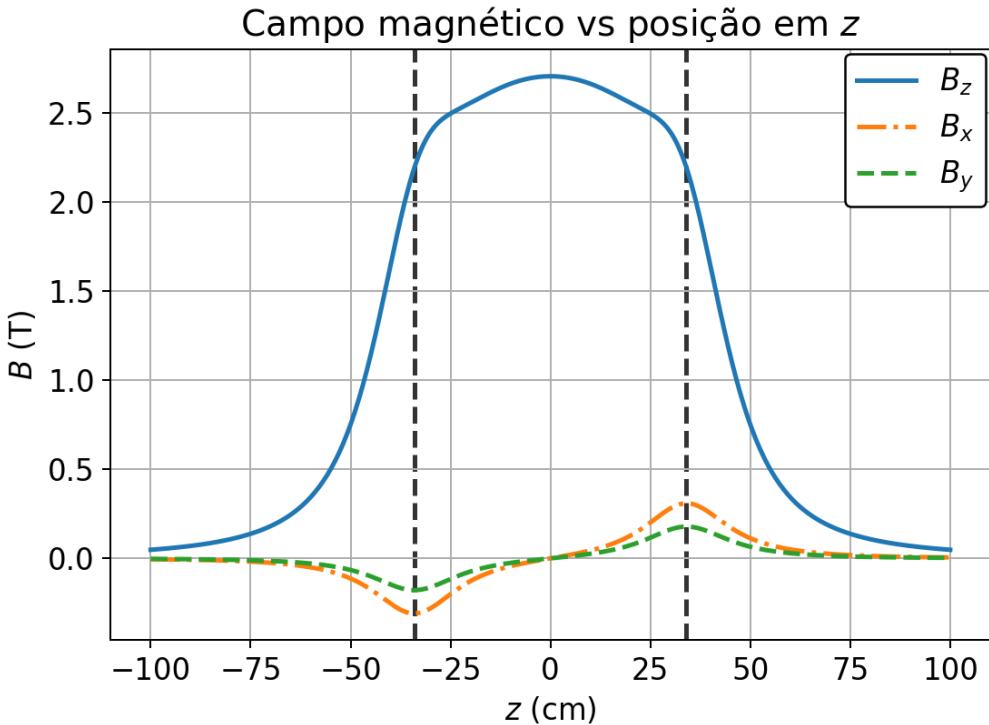


Figura 2.3: Valor do campo magnético B em tesla em função da posição em z em centímetro da bobina. A linha vertical tracejada preta indica o limite físico da bobina. O campo foi calculado à uma distância de 8 cm do eixo do solenoide. É possível ver claramente o efeito de borda que há em um solenoide finito [22].

Apesar de não ser evidente na figura 2.2, a trajetória das partículas dentro do solenoide são helicoidais devido à força de Lorentz e possuem uma determinada frequência de ciclotron [18]. Além disso, cada solenoide se comporta como uma lente grossa usada para focalizar os feixes. Em uma aproximação de um solenoide como uma lente grossa, o foco depende da rigidez magnética da partícula através da relação [16, 19]:

$$\frac{1}{f} = \frac{B_z^2}{(B\rho)^2}, \quad (2.1)$$

onde f é o ponto focal, B_z a componente z do campo magnético, e $B\rho$ é dado por:

$$B\rho = \frac{mv}{q} = \frac{\sqrt{2mE}}{q}, \quad (2.2)$$

onde E é a energia, m sua massa e q seu estado de carga.

No experimento, os campos magnéticos dos solenoides foram ajustados para focalizar o feixe de ^{17}F dentro do pAT-TPC, com energia de 51 MeV e intensidade da ordem de $10^2 \sim 10^3$ partículas por segundo. No entanto, partículas com energias e massas diferentes mas com o mesmo $B\rho$ (ou valores próximos) do feixe de interesse poderiam ser também selecionadas pelos solenoides. Isso faz com que não seja possível obter um feixe de ^{17}F com 100% de pureza, e sim um coquetel de partículas [19]. O coquetel de feixe produzido nesse experimento tinha 54% de ^{17}F , 41% de ^{16}O e cerca de 5% de ^{17}O . A figura 2.4 mostra o espectro biparamétrico de identificação de partículas obtido durante o experimento, onde é possível identificar as partículas que estão presentes no feixe (coquetel de partículas). Por fim, o feixe produzido pelo TWINSOL foi conduzido até o pAT-TPC.

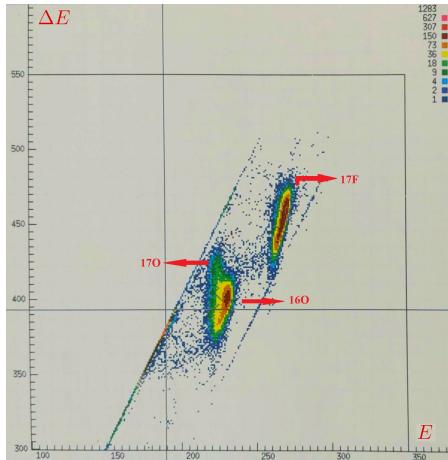


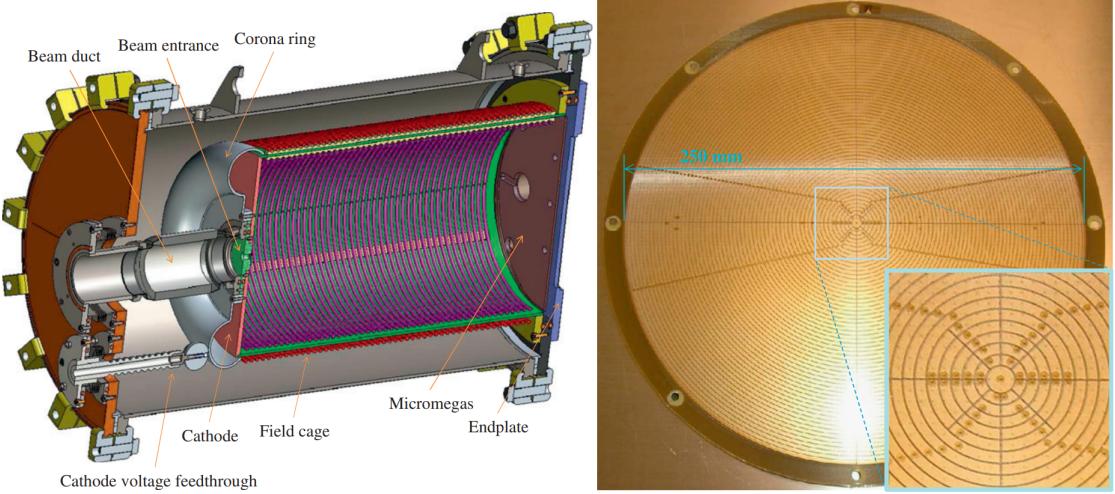
Figura 2.4: Espectro biparamétrico $\Delta E - E$ de identificação de partículas. Nele é possível identificar que o feixe possui a presença de ^{17}F , ^{16}O e uma pequena parte de ^{17}O .

Para identificar essas partículas do coquetel de feixe um sistema de detectores, formando um telescópio $\Delta E-E$, foi posicionado à 50 cm da janela do pAT-TPC, em zero graus com relação ao eixo z da câmara. Uma vez preparado o feixe, ele foi removido. O espectro biparamétrico formado pela combinação dos sinais ΔE versus $E_{\text{total}} = \Delta E + E_{\text{residual}}$, permite a identificação das partículas.

2.2 O alvo ativo pAT-TPC

O sistema de detecção utilizado nessa experiência foi o alvo ativo pAT-TPC. Alvos ativos vêm sendo cada vez mais utilizados em experimentos para investigações de reações nucleares [6], dentre os diversos laboratórios que utilizam alvos ativos podemos citar o TexAT [10] e o ACTAR [23]. A principal razão é que esses alvos funcionam também como detectores permitindo uma total identificação das partículas e de suas trajetórias dentro do alvo.

A figura 2.5a mostra o desenho esquemático do pAT-TPC, que foi o alvo ativo utilizado no experimento. O detector possui uma cela cilíndrica de 50 cm de comprimento e 28 cm de diâmetro, onde o seu eixo (direção z) é alinhado com o eixo do feixe [13], que passa pela câmara de íons no duto central. Para que o gás (utilizado como alvo) seja confinado dentro dessa câmara, utilizamos janelas finas o bastante para permitir a passagem do feixe, mas grossas o suficiente para o confinamento do gás. O feixe de ^{17}F , com energia inicial de 51 MeV, entra na câmara passando pela janela. O feixe radioativo de ^{17}F , com energia inicial de 51 MeV, perde energia durante a passagem pela janela (feita de PPTA - *p-phenylene terephthalamide* - com 12 μm de espessura) do pAT-TPC ficando com 34.76 MeV. Nesse experimento, a câmara foi preenchida com ^4He gasoso puro à uma pressão de 350 Torr que serve tanto como alvo para as reações nucleares, quanto para a própria medição e detecção dos produtos da reação [13, 24]. Tanto o feixe quanto as partículas originadas pelas reações que possam ocorrer ionizam o gás e os elétrons que surgem dessa ionização são conduzidos por um campo elétrico de 1 kV/cm perpendicular ao eixo da câmara até o plano detector (*pad plane*), o micromegas [11], localizado no final da câmara, mostrado na figura 2.5b.



(a) Visão transversal do pAT-TPC. O gás é preenchido dentro da cela que possui um campo elétrico perpendicular ao plano do micromegas, à direita da figura. O feixe incide na câmara entrando pelo duto de feixe à esquerda da figura.

(b) Foto do micromegas. O detector é multi-pixelado com uma maior densidade no centro, parte destacada na imagem. O *pad* central tem diâmetro de 5 mm enquanto que as faixas coaxiais possuem passo de 2 mm [8, 7].

Figura 2.5: Figura esquemática do pAT-TPC e o detector micromegas[13].

O micromegas é um dispositivo de amplificação de elétrons, que consiste em um plano detector com 2048 canais (pads) triangulares (ver o design do plano detector na figura 2.6 com eletrônica independente, que usa o *Generic Electronics for TPCs* (GET) [12]. Detalhes sobre a eletrônica podem ser encontrados nas Refs. [12, 7]. Cada pad do micromegas possui uma posição (x, y) fixa e a terceira coordenada z é determinada a partir do tempo de deriva dos elétrons no gás [13, 24, 8, 7]. Isso só é possível pois a velocidade de deriva (*drift*) dos elétrons é constante [25], portanto a posição em z da partícula é diretamente proporcional ao tempo de deriva. Esse princípio que deu origem ao nome de *Time Projection Chamber*, pois o evento é projetado no tempo de deriva dos elétrons no gás.

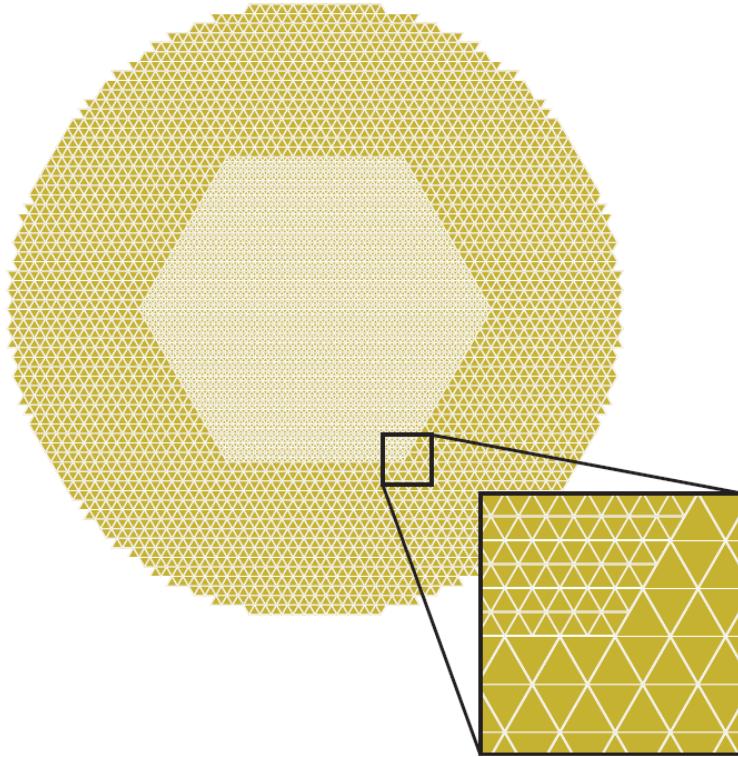


Figura 2.6: Design do plano detector. A densidade de pads é maior na região central. Com o zoom em um dos vértices da parte hexagonal é possível ver o formato triangular dos pads.

A coordenada z depende da velocidade de deriva (*drift velocity*) dos elétrons no gás. A equação 2.3 (equação de Langevin) descreve o movimento de um elétron com massa m e carga e sujeito a um campo elétrico e magnético [25]

$$m \frac{d\vec{v}}{dt} = e (\vec{E} + \vec{v} \times \vec{B}) - \frac{m}{\tau} \vec{v}, \quad (2.3)$$

onde \vec{E} é o vetor campo elétrico, \vec{B} o vetor campo magnético, \vec{v} é o vetor de velocidade do elétron e τ é o tempo de colisão médio, que depende das propriedades termodinâmicas do gás. No caso deste experimento, \vec{B} é zero e a solução estacionária para a velocidade de deriva do elétron é

$$\vec{v} = \frac{\tau}{m} e \vec{E}. \quad (2.4)$$

A velocidade de deriva depende das propriedades termodinâmicas do gás (temperatura, pressão) e também de sua condutividade elétrica [25]. Isso significa que a calibração da velocidade não depende só do campo elétrico, mas também das propriedades do gás dentro do alvo ativo [13, 25]. Para acharmos a coordenada z , basta integrar a equação 2.4 para obter

$$z = \frac{\tau}{m} e \vec{E}(t - t_0), \quad (2.5)$$

onde no tempo $t_0 = 0$ o elétron está no plano do detector ($z = 0$).

A velocidade de deriva do elétron no gás de ${}^4\text{He}$ puro deveria ser sempre constante, se as condições permanecessem as mesmas durante toda o experimento. No entanto, nas medidas desse experimento, essa velocidade variou um pouco a cada run (conjunto de eventos para serem analisados). A figura 2.7 mostra a velocidade de deriva obtida para cada run (~ 1 h) do experimento.

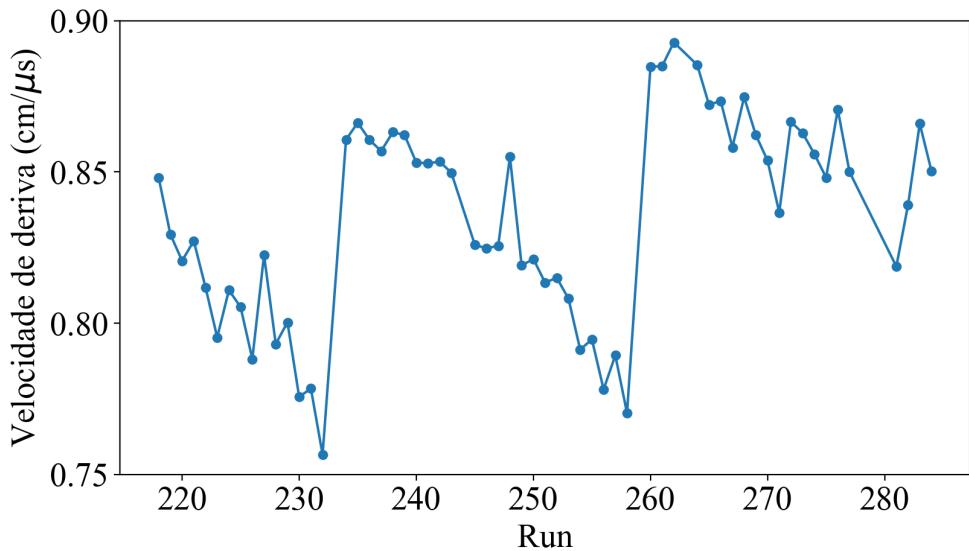


Figura 2.7: Gráfico de velocidade de deriva dos elétrons a cada run. O tempo de duração de cada run foi de aproximadamente uma hora.

Pela figura 2.7 é perceptível que a velocidade de deriva dos elétrons no gás teve uma variação por volta do 10% durante o experimento. A queda na velocidade ocorre, por exemplo, por problemas relacionados à impurezas no gás devido à filtragens de ar dentro do gás (ou *outgassing*) de alguns materiais dentro do alvo. Foi necessário corrigir os dados experimentais run por run a partir desses valores de velocidade de deriva.

Por exemplo, para o run 218, a velocidade de deriva dos elétrons foi aproximadamente de 0.85 cm/ μ s. Isso significa que os elétrons produzidos próximo na janela de entrada do detector percorrem os 50 cm da câmara em cerca de 59 μ s. Dividindo esse tempo pelos 512 canais (largura de cada bin dos pulsos gerados), tem-se que cada canal (time bucket) possui cerca de 115 ns de largura.

O pAT-TPC conta com uma camada extra de *thick gems* acoplada ao detector micromegas. *Thick gems* usam do fato de que, no momento em que o elétron passa para uma região de campo elétrico ordens de grandeza maior que de sua origem, ocorre a ionização

secundária (quando o elétron ioniza o gás). Isso provoca o que é chamado de avalanche de elétrons, amplificando a intensidade do sinal recebido [12]. A camada com *thick gems* fica por volta de $100 \sim 120 \mu\text{m}$ do pad plane, com um campo elétrico 10 vezes maior que o campo elétrico da câmara. A figura 2.8 mostra a esquematização do micromegas, onde na eletrônica de saída é produzido um sinal em função do tempo.

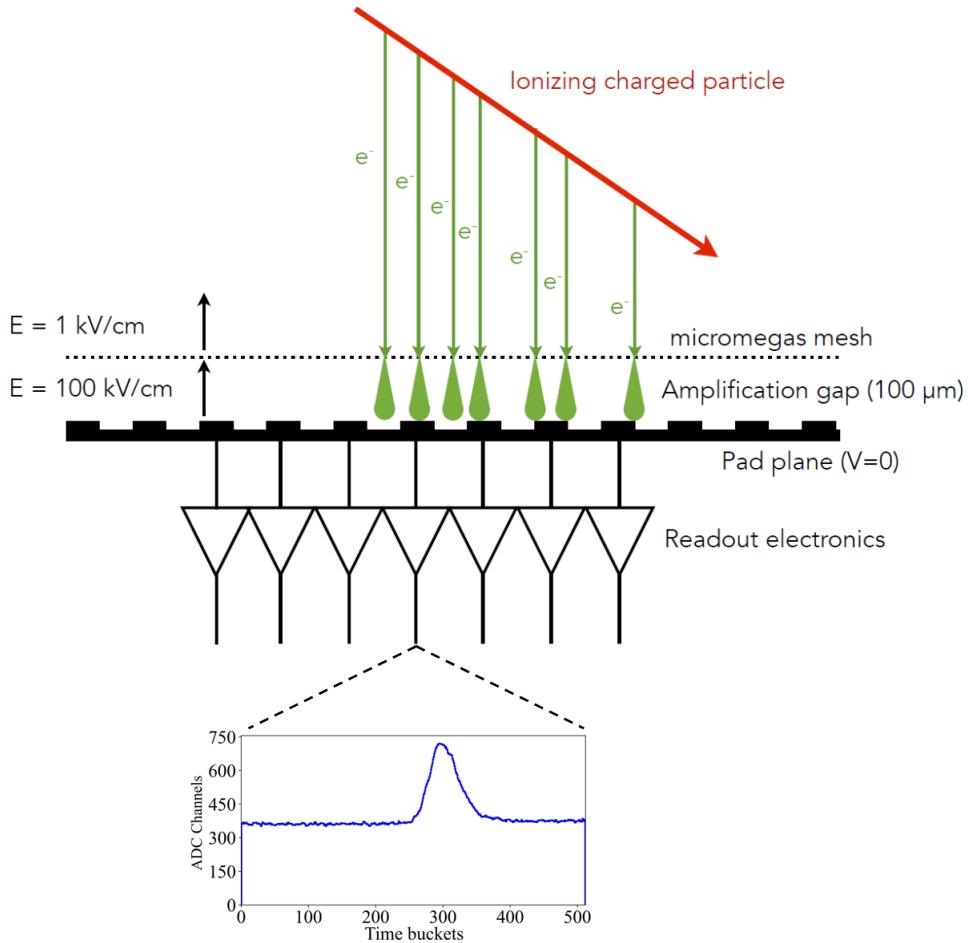


Figura 2.8: Plano do micromegas com a esquematização das *thick gems*. Os elétrons quando passam para um campo elétrico mais intenso ionizam o gás, produzindo ainda mais elétrons (evento chamado de avalanche de elétrons). Cada canal da eletrônica de saída produz um pulso como mostrado na parte de baixo da figura.

Cada interação de uma partícula carregada com o gás (liberação de elétrons por ionização do gás) é detectada como um pulso eletrônico, onde o centroide corresponde ao tempo de deriva. A carga acumulada Q dessa interação é a área do pulso associado ao centroide. Cada centroide então representa um ponto no espaço (x, y, t, Q) . Um exemplo de evento reconstruído está na figura 2.9.

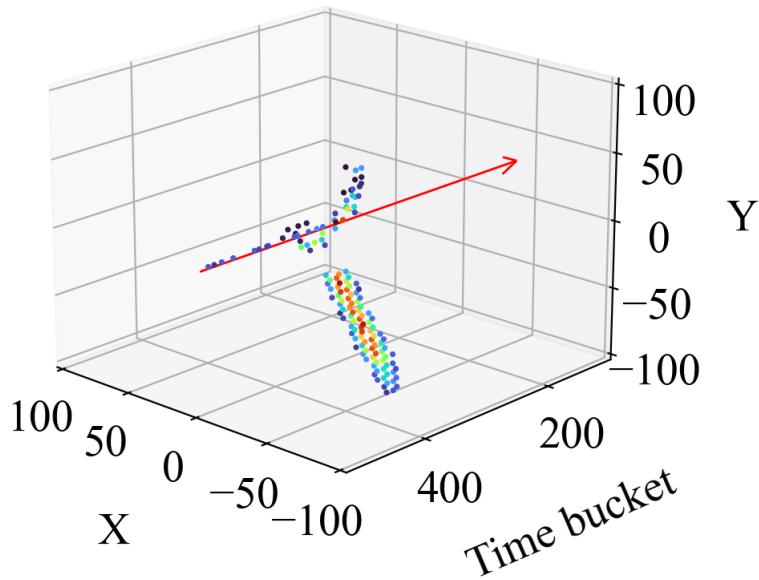


Figura 2.9: Evento reconstruído a partir da análise dos pulsos gerados pelo micromegas. A cor representa a carga integrada de cada ponto de interação com o gás.

Cada evento produz em média 300 pulsos. Para que fosse possível reconstruir as trajetórias das partículas, foi preciso analisar cada um desses 300 pulsos de todos os eventos da trajetória. No entanto, o número de evento reconstruídos é da ordem de milhões, portanto a quantidade de sinais que precisam ser analisados é de grandes proporções, em comparação com experimentos no passado em física nuclear onde apenas alguns canais de detecção são utilizados, gerando um evento por canal. É essa grande quantidade de dados também gerou a necessidade de desenvolvimento de algoritmos com alta eficiência em tempo para que a análise seja muito mais rápida. Justamente esse é um dos principais objetivos desse trabalho, que é desenvolver algoritmos de machine learning para a análise de dados com o alvo ativo. Para a análise completa do nosso experimento foram seguidas as seguintes etapas:

- Análise dos pulsos de cada interação das partículas com o gás. Isso envolve remover o fundo, localizar os picos e obter os tempos e carga integrada de cada caso;
- Reconstruir eventos (trajetória das partículas) em 3D (nuvens de pontos) a partir da análise de sinais. As nuvens de pontos precisam ser analisadas com algoritmos de reconhecimento de padrões que permitem ajustar as trajetórias das partículas em 3D;
- Reconstruir a cinemática das partículas com as trajetórias e energia depositada no gás. Isto permite obter as distribuições angulares.

A descrição completa da análise dos pulsos, reconstituição de eventos, reconstrução da cinemática, gráficos das distribuições angulares e resultados são apresentadas nos próximos capítulos.

Capítulo 3

Desenvolvimento de ferramentas de *machine learning* para análise de dados

Uma das possibilidades para que possamos analisar a grande quantidade de dados gerados em experimentos com alvo ativo é a utilização de técnicas de machine learning. Nesse capítulo explicamos a metodologia usada para a implementação dessas ferramentas para a análise dos dados obtidos na experiência realizada com $^{17}\text{F} + ^4\text{He}$. O apêndice A mostra algumas aplicações recentes de machine learning na física nuclear.

De modo geral, machine learning é a utilização de algoritmos para extrair informações de uma grande quantidade de dados brutos e representá-los através de algum tipo de modelo matemático. Machine learning é, portanto, uma técnica onde os algoritmos analisam os dados aprendendo (ou sendo ensinado) com eles, sem que tivessem sido explicitamente programados para isso [26]. A ideia de machine learning não é nova e surgiu na década de 50 [27, 26]. No entanto, somente nos dias de hoje, com a possibilidade de processamento paralelos de CPU é que ela pode ser aplicada. Essa ideia de processamento foi também inspirada no funcionamento da rede neural biológica, onde os neurônios interagem enviando sinais através de conexões que se desenvolvem com treinamento. Aqui as interações seriam os sinais na forma de funções matemáticas enviadas ou trocadas entre as camadas, que poderiam ser associados aos neurônios. Aqui então, uma rede neural seria uma sucessão de camadas interligadas por operações e funções matemáticas. Isso inspirou o uso do modelo matemático simples de uma função linear nos parâmetros para um neurônio artificial ou camada [27]:

$$y = f \left(\sum_{i=1}^n \omega_i x_i + b_i \right) = f(z), \quad (3.1)$$

onde y é o resultado da função matemática da interação entre os neurônios (camadas),

dada pela função de ativação f . Essa função f é dada pela soma ponderada da variável de entrada x_i (fornecida pelos outros n neurônios) com um peso w_i . O termo b_i corresponde ao parâmetro bias. A ideia é fazer um neurônio receber a informação de todos os outros neurônios da camada anterior, fazendo uma média ponderada (onde o peso que será estimado pelo algoritmo de machine learning) e somando com um termo independente (bias, que também é estimado). Os parâmetros ω_i e b_i serão estimados através de um determinado procedimento, chamado de minimização (ou treino da rede neural).

3.1 Tipos de redes neurais

Como mencionamos, a ideia do machine learning é baseado na ideia da utilização de redes neurais. Uma rede neural artificial, *Artificial Neural Network* (ANN), é um modelo computacional que consiste de camadas de neurônios. ANNs foram desenvolvidas para o estudo de inteligência artificial [26, 28]. ANNs consistem principalmente numa camada de entrada (*input layer*), uma camada de saída (*output layer*) e eventuais camadas entre essas duas, chamadas de camadas ocultas (*hidden layers*). Os tipos mais comuns de redes neurais aplicadas a análise de dados são:

Feed-Forward Neural Networks

A *Feed-Forward Neural Networks* (FFNN) é a primeira e mais simples rede neural desenvolvida [29, 30]. Nessa rede a informação se move apenas para frente através de camadas (da camada de entrada até a camada de saída). A figura 3.1 mostra uma representação de rede, onde os neurônios são representados por círculos, enquanto que as linhas mostram as conexões entre os neurônios. Cada neurônio recebe informação de todos os neurônios da camada anterior, portanto a rede é chamada de totalmente conectada, fully-connected (FC), FFNN.

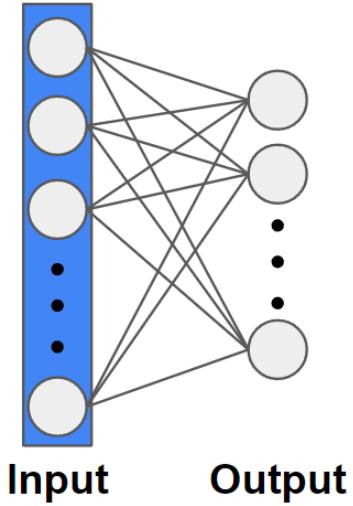


Figura 3.1: Exemplo de FFNN. A camada de entrada na esquerda propaga a informação para a direita (camada de saída). Todos os neurônios entre camadas estão conectados entre si.

Convolutional Neural Network

Uma variante da FFNN é a chamada de rede neural convolucional, *Convolutional Neural Network* (CNN). Essa rede consiste em utilizar duas funções convoluídas.

A convolução de uma função $f(t)$ por uma função $g(t)$ é dada por $(f * g)(t)$ e definida como:

$$(f * g)(t) \equiv \int_{-\infty}^{\infty} f(\tau)g(t - \tau)d\tau. \quad (3.2)$$

Para o caso discreto, com g sendo uma função resposta finita de tamanho $2M$ (M é um valor inteiro maior que zero), temos

$$(f * g)[n] = \sum_{m=-M}^{M} f[n - m]g[m]. \quad (3.3)$$

Convoluções são invariantes sobre operações de rotação e translação, portanto são muito utilizadas para processamento de sinais e imagens [31]. Além disso, a convolução pode ser aplicada de forma discretizada sobre um vetor, se transformando numa forma de filtro. Para ilustrar o que significa isso, no caso discreto e unidimensional, a figura 3.2 mostra o processo de convolução de um vetor de tamanho 9 com um filtro de tamanho 3.

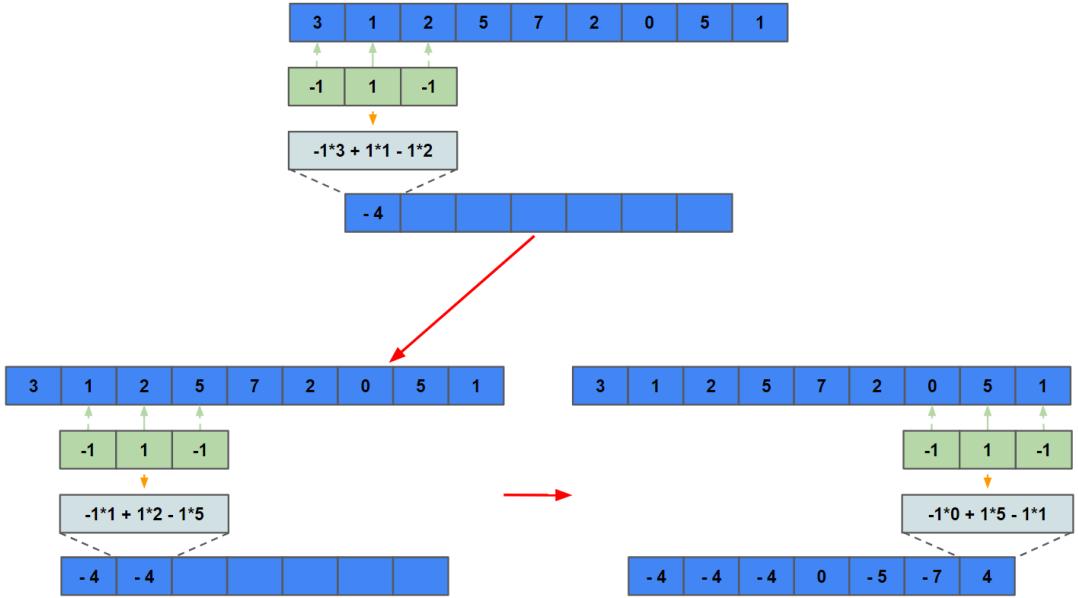


Figura 3.2: Processo de convolução entre sinal azul em cima e o filtro em verde, resultando no sinal azul embaixo. A multiplicação é feita ponto a ponto e está indicada na caixa azul-clara.

Percebe-se que o sinal resultante tem dimensão menor que o sinal original. O filtro (também chamado de *kernel*) atua em pontos que possuam vizinhos tais que o filtro possa fazer a multiplicação ponto a ponto. Esse tipo de convolução tem o que chamamos de emparelhamento válido ou em inglês “*valid padding*”. O tamanho n_2 resultante do vetor de saída é

$$n_2 = n_1 - m + 1, \quad (3.4)$$

onde n_1 é o tamanho do vetor de entrada e m o tamanho do filtro (*kernel size*). Uma alternativa para que o vetor de saída tenha o mesmo tamanho do vetor de entrada é acrescentar zeros em torno da entrada, de forma que o vetor de saída tenha o mesmo tamanho que o vetor de entrada. Nesse caso dizemos que o emparelhamento é igual ou “*same padding*”. A figura 3.3 ilustra esse processo.

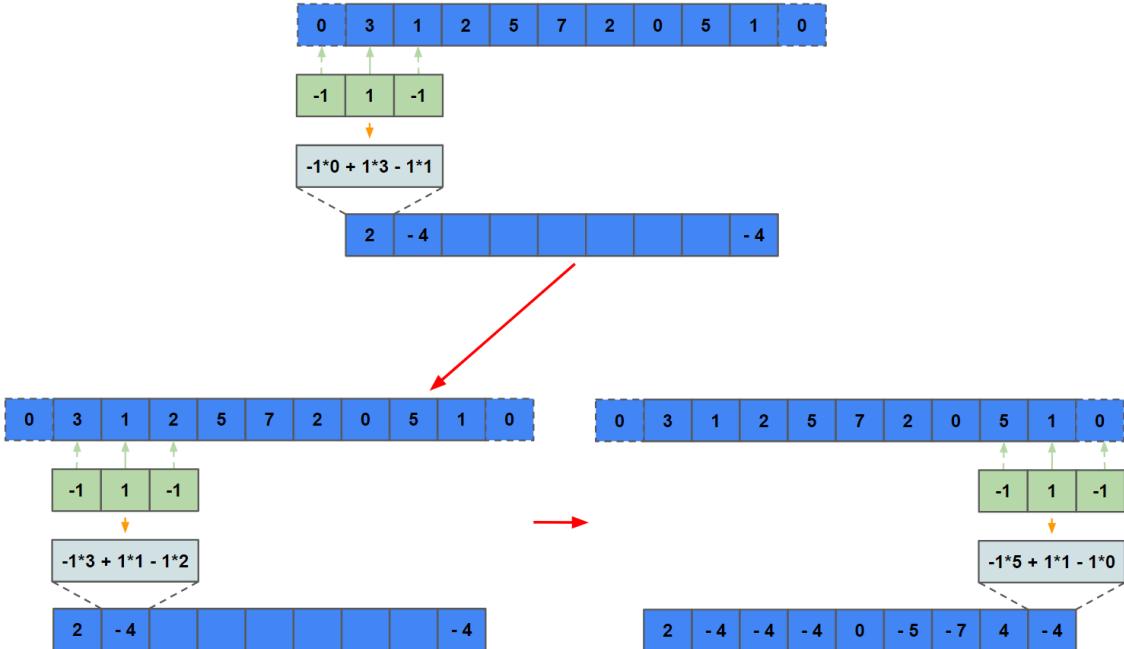


Figura 3.3: Processo de convolução entre o sinal azul em cima e o filtro em verde, resultando no sinal azul embaixo. Agora são acrescentados zeros no inicio e no final do vetor para que o vetor saída tenha o mesmo tamanho do vetor de entrada (nesse caso 9).

No contexto de machine learning, não sabemos, a priori, quais os valores dos filtros devem ser utilizados. Temos informações apenas de seus tamanhos e como devem agir. No entanto, podemos estimar os valores dos filtros através do treino da rede neural, para obter o resultado final. Esse treino fornece então os valores dos filtros a serem aplicados que forneçam o resultado esperado.

Cada filtro aplicado gera um mapa característico (*feature map*), que é o resultado da atuação do filtro em um vetor. Usualmente, em uma CNN se escolhe o tamanho do filtro, *padding* (*valid* ou *same*) e quantos filtros serão aplicados (para saber quantos mapas característicos serão gerados). Como temos vários mapas gerados por cada filtro, isso acarreta em um aumento de dimensionalidade. Para selecionar (ou filtrar) os mapas são utilizados critérios, como por exemplo o de selecionar valores máximos dos mapas gerados dada uma janela de atuação, ou seja, de quantos mapas serão comparados para a seleção do valor máximo. O Max-Pooling¹ faz isso, selecionando valores máximos para uma determinada quantidade de mapas sendo comparados (*pool size*).

Existem disponíveis na literatura várias outras arquiteturas de redes neurais que não serão discutidas aqui. Uma descrição dessas arquiteturas podem ser encontradas nas referências [32, 33].

¹O Max-Pooling máximo é um processo de discretização baseado em amostras. O objetivo é reduzir a amostragem de uma representação de entrada, selecionando amostras pelos seus valores máximos.

3.2 Estrutura da rede neural

Em nosso trabalho vamos utilizar uma rede neural supervisionada, que será discutida em mais detalhes nas próximas seções. Mas de uma forma geral, para a construção de uma rede neural é preciso definir sua estrutura. A estrutura da rede neural possui camadas e cada camada pode possuir uma função de ativação. No geral, tanto a camada de entrada quanto a de saída possuem dimensão fixa. Para cada camada da arquitetura devemos escolher sua função de ativação que dependem dos parâmetros escolhidos. Tanto FFNNs quanto CNNs podem possuir funções de ativação (função f da equação 3.1) para melhorar o treino com dados que não possuem uma dependência linear. Dentre muitas funções de ativação podemos citar a *Rectified Linear Units* (ReLU)[34], sigmoide [35], linear e tangente hiperbólica [36]. A figura 3.4 mostra os gráficos dessas funções de ativação, onde no eixo x é o argumento e no eixo y o resultado da função.

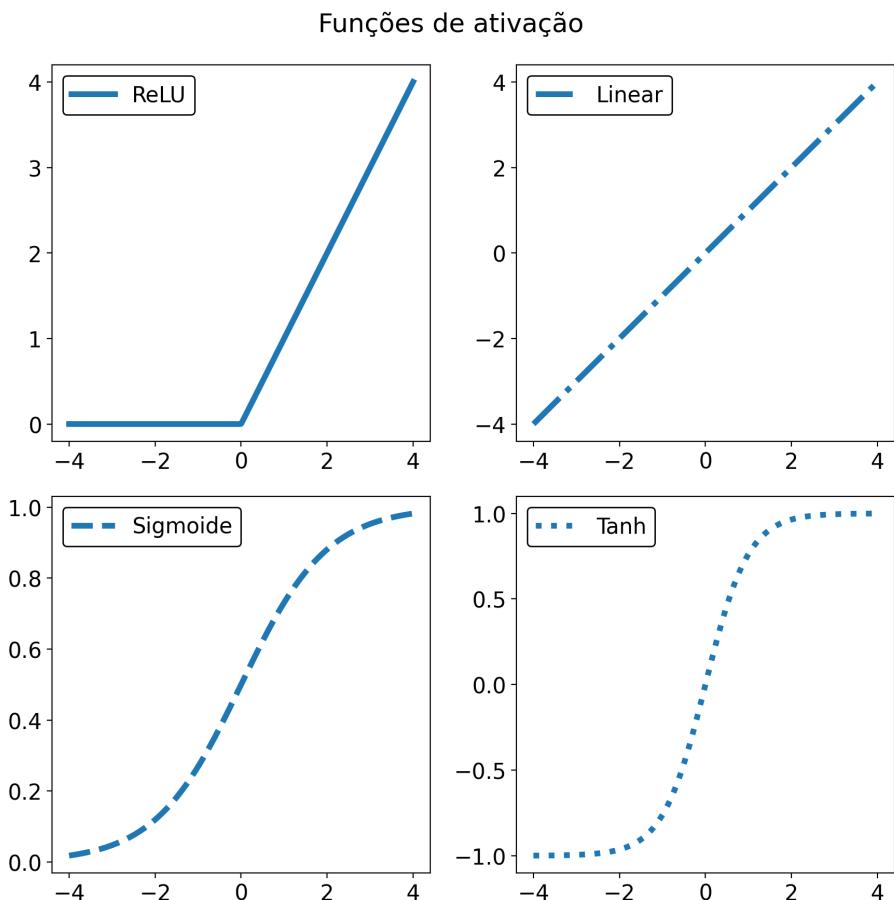


Figura 3.4: Funções de ativação e seus respectivos gráficos.

O próximo passo é definir funções corretivas chamadas função custo (algumas vezes também chamada de função *loss*) e o otimizador. A função custo tem o papel de retornar

valores altos para previsões erradas e valores baixos para previsões corretas. Por exemplo, se queremos treinar uma rede neural para classificação binária (que prevê duas saídas possíveis), devemos usar a função custo chamada de *binary cross-entropy* dada por [37]

$$C(p(y_i)) = -\frac{1}{N} \sum_{i=1}^N y_i \log(p(y_i)) + (1 - y_i) \log(1 - p(y_i)), \quad (3.5)$$

onde y_i é o rótulo (*label*), $p(y_i)$ é a probabilidade do ponto y_i ser 1 e N é o número de pontos. O objetivo da rede neural é achar o mínimo da função $C(p(y_i))$, o que implica diretamente na melhor solução para o conjunto de dados. Isso é feito pelo método de retropropagação do erro (*backpropagation* [38]) por um otimizador. Outros exemplos de função custo são o erro quadrático médio ou *categorical cross-entropy* [39].

O otimizador, por sua vez, tem o objetivo de otimizar os parâmetros presentes na rede neural, buscando o mínimo global da função custo, o que nem sempre acontece, pois a minimização pode parar em um mínimo local da função. Existem diversos otimizadores disponíveis na literatura, como por exemplo o *Stochastic Gradient Descent* (SGD), *Adaptive Moment Estimation* (ADAM), ADAMAX [40], entre outros [41]. Por exemplo, para o SGD, temos que a atualização de parâmetros é dada por

$$\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} C(\theta), \quad (3.6)$$

onde θ_j é o parâmetro a ser atualizado, α é a taxa de aprendizado (*learning rate*) e $C(\theta)$ é a função custo que depende dos parâmetros θ .

Para enfim treinar a rede neural, devemos escolher o tamanho das amostras (*batch size*) que será usada para o treino, por iteração em cada rodada de treino. Por exemplo, se usamos 1000 dados para o treino, e o *batch size* é 500, cada rodada de treino terá duas iterações. No geral, usamos *batch sizes* pequenos, pois o consumo de memória é mais eficiente.

Para avaliação da rede neural devemos ainda utilizar dados de validação, que servem para verificar o comportamento da rede neural que está sendo treinada. Esse conjunto de dados usados não é usado para o treino, são usados apenas para verificar possíveis problemas como o *overfit*. O *overfit* ocorre quando a rede neural começa a se adequar perfeitamente aos dados de treino, perdendo a capacidade de previsão em dados que não estão sendo usados no treino pela rede neural.

Além dos dados de validação, podemos escolher métricas que auxiliam a visualização do comportamento da rede neural durante o treino e nos retornam informações importantes sobre sua qualidade. Exemplos importantes de métricas são: acurácia binária, erro

médio absoluto e acurácia categórica [42]. Por exemplo, caso seja necessário verificar se uma rede neural está fazendo previsões certas em um problema cuja classificação é binária, então a métrica deve ser a acurácia binária. Tudo depende do objetivo da rede neural. Em nosso trabalho utilizamos o erro médio absoluto e acurácia binária.

3.3 Sistemas de *machine learning*

As redes neurais e os procedimentos de validação e treino formam os sistemas de machine learning. Podemos dividir os sistemas de machine learning em quatro tipos:

Aprendizado Supervisionado

Aprendizado supervisionado é quando fornecemos para a rede neural um conjunto de dados para o treino com a solução desejada (chamados de *labels*). Um uso típico desse tipo de sistema é para problemas de classificação de imagens (identificação de figuras), previsão de valores numéricos etc [26]. Exemplos de algoritmos supervisionados são:

- *k-Nearest Neighbors* (é necessário saber o número de aglomerados a priori) [43]
- Regressão linear
- *Support Vector Machines* (SVMs)
- *Decision Trees e Random Forests*
- Redes neurais

Aprendizado não supervisionado

Aprendizado não supervisionado é quando fornecemos o conjunto de dados sem rótulos ou características a priori. Nesse caso, o sistema deve aprender a realizar uma determinada tarefa sem supervisão. Um problema comum, por exemplo, é quando queremos identificar aglomerados em um conjunto de dados (*clustering*) [44, 45], ou remover ruído com estruturas chamadas *autoencoder*¹ [46].

¹*Autoencoder* é uma técnica de aprendizado não supervisionado para redes neurais aprenderem a compressir e codificar dados de modo eficiente.

Aprendizado semi supervisionado

Temos ainda o aprendizado semi supervisionado, quando apenas parte do conjunto de dados para o treino possui *labels*. Isso é comum quando se obtém conjuntos de dados diferentes e apenas parte deles foi classificado [47].

Aprendizado por reforço

Aprendizado por reforço é quando um sistema, chamado de *agente*, aprende através do ambiente, realizando ações que maximizam sua recompensa. Por exemplo, caso o sistema realize uma ação incorreta, ele recebe uma penalidade, fazendo com que procure outra maneira de realizar a ação, se essa maneira for a correta o sistema é recompensado [48]. Esse tipo de sistema é muito usado, por exemplo, em automatização robótica, como carros que pilotam sozinhos, robôs que aprendem a andar etc. [49]

Alguns usos recentes de machine learning em física nuclear são apresentados no apêndice A. Em nosso trabalho vamos utilizar técnicas de machine learning para analisar dados de reações nucleares obtidos de medidas com um instrumento específico, alvo ativo.

Capítulo 4

Reconstrução de nuvens de pontos a partir de algoritmos de *machine learning*

Nesse capítulo vamos descrever o procedimento usado para criar as nuvens de pontos (*point clouds*) a partir dos pulsos medidos em cada pixel do micromegas, usando algoritmos de machine learning supervisionado.

Algoritmos baseados em CNNs vem sendo usados com relativo sucesso para o processamento e análise de sinais [6], como por exemplo para discriminação de pulsos [50]. CNNs possuem a capacidade de fazer ajustes multidimensionais e aprender padrões complexos. Além disso, CNNs são mais eficientes em tempo para a análise de grandes quantidades de dados em comparação com algoritmos comuns [6].

Para que possamos obter dados de posição, trajetória e energia das partículas nas medidas com alvo ativo precisamos realizar análise dos pulsos medidos. A quantidade de pulsos gerados nesse experimento é da ordem de centenas de milhões. Para a análise completa dos pulsos, são necessárias diferentes etapas que envolvem processos que muitas vezes precisam ser refeitos por causa de algum erro no processo análise. Notar algum erro no processo, ou ter que mudar algum parâmetro da análise, significaria ter que analisar novamente os dados, sendo um processo muito custoso em tempo.

O uso de algoritmos de CNNs tem o objetivo de diminuir o tempo consumido para a análise dos pulsos. As redes neurais criadas são treinadas uma única vez e podem ser utilizadas de modo separado e/ou acoplado [6], como está mostrado na sequencia do capítulo.

Os pulsos medidos com o detector micromegas são digitalizados em 512 intervalos de tempo (*time buckets*). A análise desses espectros temporais (evolução do pulso

com o tempo) foi dividida em três etapas: correção do fundo, deconvolução do sinal e identificação de picos. As redes neurais criadas foram supervisionadas, o que significa que foi necessário incluir um grande volume de dados de entrada para o algoritmo aprender a lógica da operação. Para isso foi feito um grande banco de dados, utilizado no treinamento das redes neurais, que se beneficiam da grande quantidade de dados usada para o treino [26]. Na seção 4.1, está mostrado como o banco de dados para o treino das redes neurais desenvolvidas foi criado, e a construção das redes neurais está na seção 4.2.

4.1 Construção do banco de dados para as redes neurais

Cada um dos pixels do detector micromegas possui uma eletrônica independente que gera pulsos digitais como os mostrados na 4.1. Centenas de sinais como esses são produzidas em cada um dos eventos.

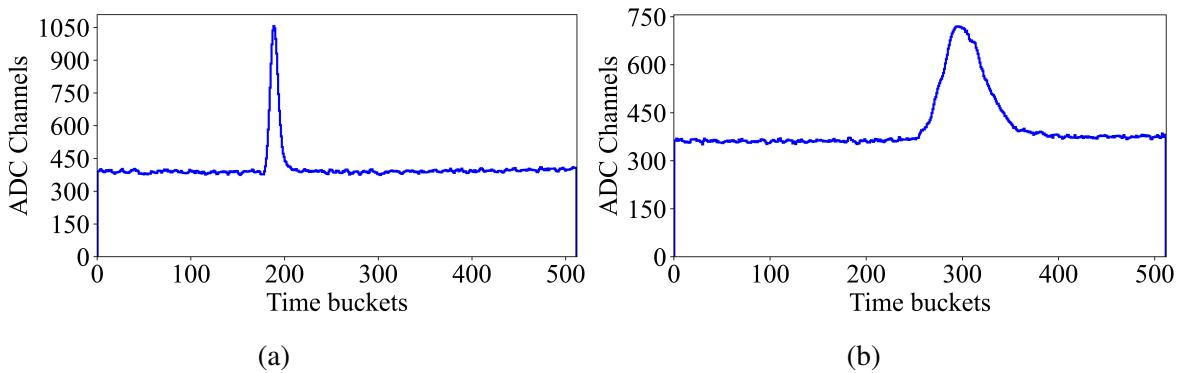


Figura 4.1: Exemplos de sinais produzidos pelos canais do detector. Em 4.1a o sinal possui apenas um pulso, enquanto em 4.1b há vários pulsos em sobreposição, formando um único pulso com largura maior que em 4.1a.

Na figura 4.1, o eixo x corresponde à cada um dos 512 *time buckets* possui largura de cerca de 115 ns. No eixo y tem-se o canal de ADC para cada time bucket. Na figura 4.1a há um sinal com um pedestal (fundo ou *baseline*) com altura entre 300 e 450 canais, e um pulso estreito em cima. O fundo está composto por ruído de alta e baixa frequência, o que geralmente dificulta o ajuste com uma função analítica. Como mostrado na seção 2, os elétrons que surgiram da ionização do gás foram conduzidos perpendicularmente pelo campo elétrico até o detector. A interação da partícula com o gás é evidenciada justamente pelo pulso presente em 4.1a. O centroide de cada pulso está relacionado ao tempo de deriva dos elétrons (em unidade de time buckets) que é usado para determinar a coordenada z da interação da partícula com o gás. A energia depositada no gás é obtida a partir da área do pulso sem fundo.

Para a figura 4.1a tem-se apenas um pulso estreito, o que corresponde à uma partícula incidindo paralelamente ao plano do detector (perpendicular ao campo elétrico). No caso da figura 4.1b, há uma distribuição ampla do sinal do tempo, o que corresponde à uma partícula incidindo quase perpendicularmente ao plano do detector. A ilustração desse processo está na figura 4.2, que mostra que o sinal resposta do processo de passagem de uma partícula carregada no gás depende do ângulo relativo ao plano de projeção. Trajetórias paralelas ao plano de detecção tem distribuição em tempo bem definidas pois a interação corresponde a apenas um ponto no eixo de projeção. Trajetórias perpendiculares ao plano de detecção tem distribuições em tempo mais largas devido a que os elétrons são produzidos numa região mais ampla.

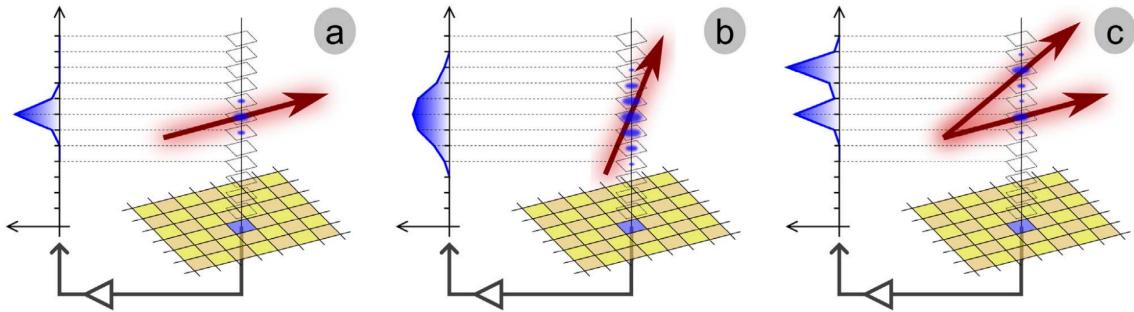


Figura 4.2: Ilustração que mostra a variação no formato da carga coletada a partir da passagem de uma partícula carregada dentro do TPC, onde o plano do detector está embaixo. No lado esquerdo de cada imagem, a distribuição do sinal coletado por um único pad (escuro) do plano de coleta é mostrado (o canal eletrônico de leitura é representado pela seta cinza em negrito). No caso de uma trajetória quase horizontal em relação ao plano do detector (a), o sinal é uma distribuição estreita, enquanto para uma trajetória próxima a uma direção vertical (ou perpendicular) em relação ao detector (b), a distribuição deve ser muito mais ampla (vários pontos de interação da partícula com o gás devem ser extraídos desse sinal). A última imagem ilustra o caso em mais de uma trajetória de partículas contribui para o sinal [12].

Para analisar os pulsos deve-se primeiro remover o fundo (pedestal ou baseline) dos sinais. O sinal de fundo é complexo e pode variar em função do canal e também por evento. O sinal pode ser alterado por flutuações causadas pelo circuito eletrônico e por efeitos sistemáticos gerados pela memória de buffer circular, podendo o tornar não analítico [6, 12]. Após a eliminação do fundo, os sinais foram tratados realizando uma deconvolução, aumentando a resolução dos pico em várias gaussianas com centróides que correspondem a informação temporal do sinal. Essas gaussianas são então usadas para a reconstrução da nuvem de pontos da trajetória das partículas no alvo gasoso. Todo esse processo foi realizado nesse trabalho com algoritmos de machine learning supervisionado que desenvolvemos [6]. Para isso, foi criado um banco de dados que serviu de saída e/ou

entrada para o treino das redes neurais. O processo de criação dos dados para o treino das redes neurais está descrito na seção 4.1.1, enquanto o processo de criação das redes neurais é descrito na seção 4.2.

4.1.1 Estimativa do fundo

Para que pudéssemos realizar a análise dos pulsos temporais, era necessário a eliminação do fundo. A primeira tentativa para estimarmos o sinal sem o fundo foi desenvolvida utilizando-se uma transformada de Fourier e um filtro passa-baixa, que nada mais é do que a função resposta do detector fornecida na Ref. [12]. Para explicarmos o procedimento, vamos primeiramente considerar uma função $f(t)$ qualquer, sua transformada de Fourier é dada por

$$\hat{f}(\nu) = \mathcal{F}[f(t)] = \int_{-\infty}^{\infty} f(t)e^{-2\pi i \nu t} dt. \quad (4.1)$$

Primeiro calculamos a transformada de Fourier $\hat{f}(\nu)$ do sinal, em seguida o multiplicamos pela função resposta do detector $h(\nu)$ dada por [12]

$$h(\nu) = A * \exp(\nu\tau) (\nu\tau)^3 \sin(\nu\tau), \quad (4.2)$$

onde A está relacionado com o ganho de amplificação e τ é o tempo de pico (peaking time), que é o tempo de modelagem da cadeia de amplificação [12]. Do teorema da convolução, temos que [51]

$$\mathcal{F}^{-1}[\hat{f}(\nu)\hat{g}(\nu)] = (f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t - \tau)d\tau, \quad (4.3)$$

onde $(f * g)(t)$ é a convolução entre $f(t)$ e $g(t)$. Multiplicar o sinal transformado por $h(\nu)$ e depois inverter a transformação é o mesmo que convoluir o sinal original com a transformação inversa de $h(\nu)$, o que resulta no sinal sem o fundo [7, 12]. Resultados desse procedimento estão na figura 4.3.

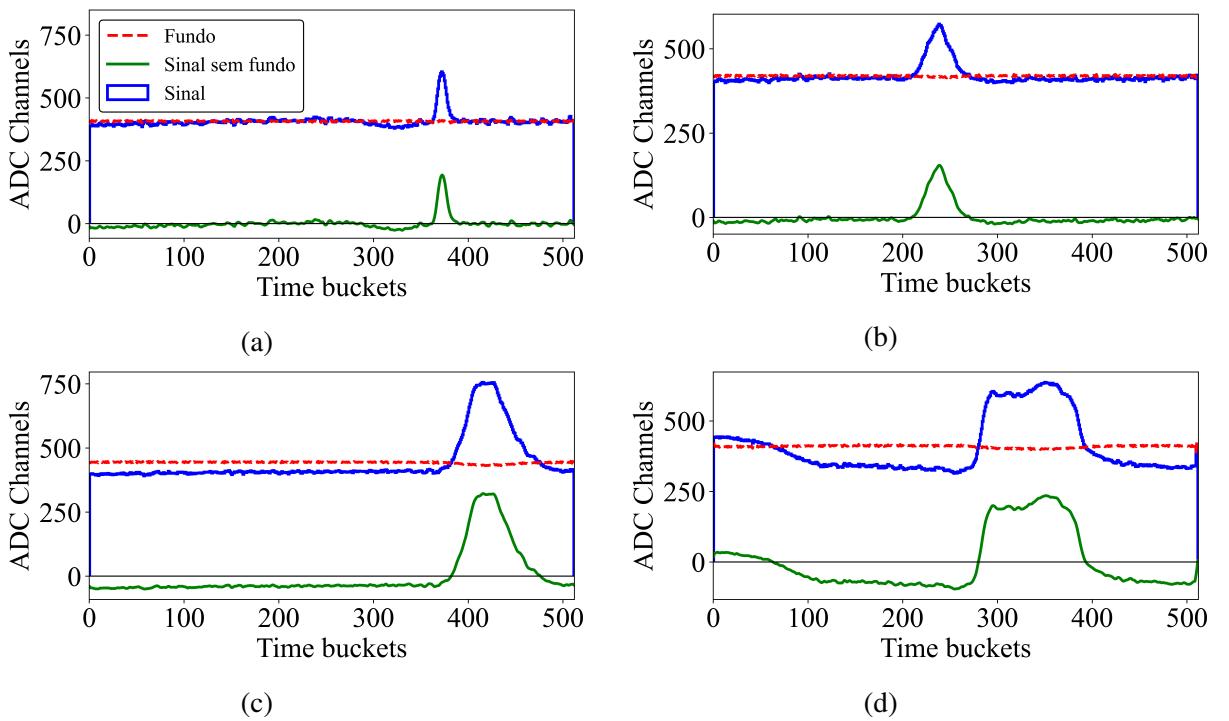


Figura 4.3: Histogramas com as respectivas baselines (linhas tracejadas) estimadas pelo método da convolução. O espectro resultante (sem o fundo) está em verde.

Fica claro que visualmente, por exemplo na figura 4.3d, que o filtro utilizado não é a melhor função resposta do detector. Poderia-se estimar essa função resposta empiricamente, porém os canais auxiliares chamados de *Fixed Pattern Noise* (FPN) [12] usados para esta estimativa não foram armazenados. Portanto, a estimativa do fundo foi feita sinal por sinal [6, 12]. Para isso, o fundo foi determinado usando o algoritmo BACKGROUND REMOVAL da biblioteca TSpectrum do ROOT [52]. A função tem a capacidade de separar o fundo dos picos presentes no espectro [53, 54, 55]. Exemplos de estimativa do fundo estão na figura 4.4.

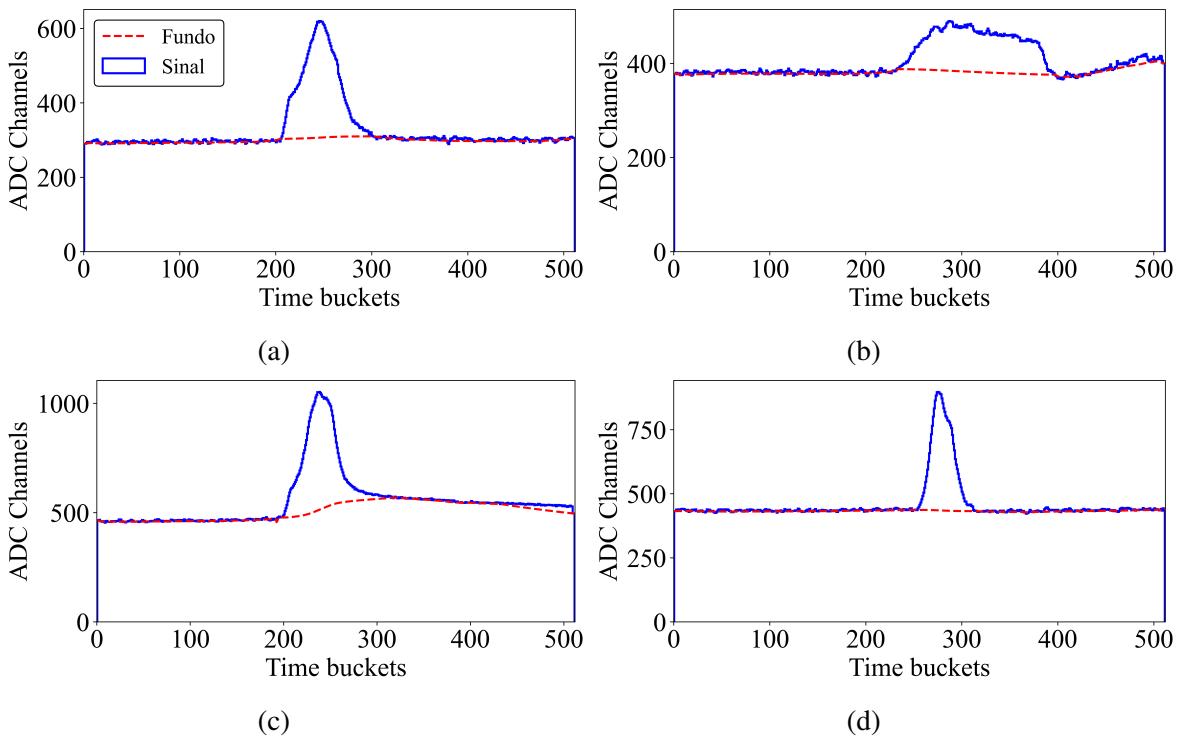


Figura 4.4: Histogramas com as respectivas baselines (linhas tracejadas) calculadas pelo TSpectrum.

Os resultados dos cálculos para o ruído de fundo de cada sinal foram armazenados e usados na etapa seguinte de análise dos mesmos. Nessa próxima etapa a deconvolução dos sinais é realizada com os sinais tendo o ruído de fundo devidamente subtraído. Para diminuir as possíveis flutuações numéricas após a remoção do fundo, o valor mínimo do sinal sem o fundo deve ser zero. Uma vez eliminado o fundo, o código busca e identifica os picos e as correspondentes cargas do sinal. Para os picos compostos é então realizada a deconvolução, conforme descrita na seção seguinte.

4.1.2 Deconvolução do sinal

Dependendo do ângulo da trajetória com respeito ao plano de detecção, a largura dos picos com cada pixel pode mudar (figura 4.2). Em termos gerais, os picos podem ser descritos como uma superposição de Gaussianas com uma largura fixa que é definida pelo eletrônica do alvo ativo. Precisamos então deconvolver os picos em suas componentes. Para tanto utilizamos o algoritmo GOLD DECONVOLUTION presente na biblioteca TSpectrum do ROOT [56]. O algoritmo tem como objetivo fazer a deconvolução do espetro, gerando uma função (nesse caso um pulso gaussiano) resposta de acordo com o desvio padrão esperado para os pulsos. O sinal resposta corresponde ao espetro com as

gaussianas não sobrepõem. Isso significa que foi necessário determinar qual o valor da largura dos pulsos para buscar a função resposta. A largura dos pulsos foi determinada fazendo a análise de sinais que possuem apenas 1 pico, fazendo um ajuste pelo método dos mínimos quadrados (MMQ) de uma gaussiana. Para buscar espectros com apenas um pico, foi usado o algoritmo de identificação de picos PEAK_FINDER, presente na biblioteca do `scipy` [57], e para o ajuste da gaussiana foi usada o pacote LMFIT [58]. O valor do desvio padrão encontrado foi de 4.09 (17) *time buckets*. A largura escolhida foi ligeiramente maior pois, verificando empiricamente, em alguns casos o algoritmo separava o que deveria ser uma única gaussiana em duas. O valor do desvio padrão usado na deconvolução foi de 4.30 *time buckets*. Foi determinado também o número de iterações do algoritmo de deconvolução. O número de iterações escolhido foi de 700, menos que isso o algoritmo não estava separando totalmente picos sobrepostos. O limiar para a escolha de um ponto como um pico foi definido como ter altura maior que 20% do valor máximo do sinal. Resultados da deconvolução estão na figura 4.5.

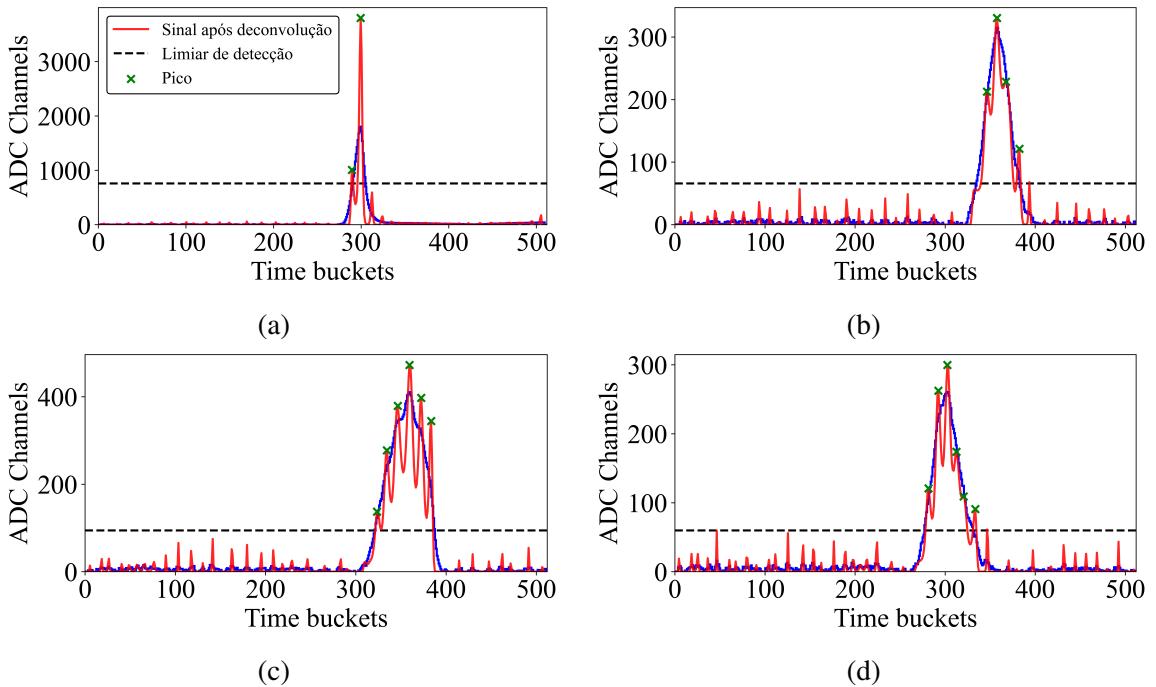


Figura 4.5: Histogramas sem as *baselines* antes (em azul) e depois da deconvolução (em vermelho). Os picos (em verde) e o limiar (linha tracejada preta) de identificação também estão indicados.

O algoritmo de deconvolução também determina a posição dos centróides encontrados, que indica a localização de um pico. Os mesmos picos podem ser obtidos com o PEAK_FINDER, com a vantagem de que o algoritmo possui muitos parâmetros diferentes para calibração, melhorando a identificação em comparação com os picos identifi-

cados pelo algoritmo do TSpectrum. A execução de 200.000 sinais, desde a estimativa e remoção do fundo, até a identificação dos centróides, demora cerca de 23.25 minutos, usando o processador Ryzen 5 3600X.

Para determinar a carga acumulada Q de cada ponto, é necessário calcular a área do centróide do pico identificado. A área do sinal antes e depois da deconvolução é a mesma, portanto pode-se analisar diretamente o sinal após a deconvolução. Para determinar a área do pulso consideramos a integral simples gaussiana usando a largura de cada pico após a deconvolução. O desvio padrão (largura) dos picos após a deconvolução foi $\sigma_{dd} = 1.1543$ (44) *time buckets*. Considerando esse desvio, a carga integrada para cada ponto é dada por:

$$Q = \int_{-\infty}^{\infty} Ae^{-(t' - t_i)^2 / 2\sigma_{dd}^2} dt' = A |\sigma_{dd}| \sqrt{2\pi}, \quad (4.4)$$

onde A é a amplitude do ponto com centróide t_i e desvio padrão após a deconvolução σ_{dd} . Com o banco de dados para a deconvolução e também para a identificação de picos, basta criar as redes neurais, que serão descritas na seção 4.2.

4.2 Análise dos pulsos com *machine learning*

Com machine learning tem-se a possibilidade de criar algoritmos de alta complexidade sem definir operações explícitas. Usando os resultados das seções anteriores foram desenvolvidas três redes neurais, com o objetivo de: estimar o fundo (subseção 4.2.1), fazer a deconvolução (4.2.2) e por fim identificar os picos (subseção 4.2.3).

4.2.1 Rede neural para o fundo

O objetivo foi criar uma rede neural que reproduza o comportamento do algoritmo BACKGROUND REMOVAL que estima o fundo do sinal, discutido na seção 4.1.1. A rede neural é supervisionada, onde os dados de entrada são os sinais brutos e as saídas devem ser os fundos de cada sinal. A arquitetura dessa rede neural é apresentada esquematicamente na figura 4.6.

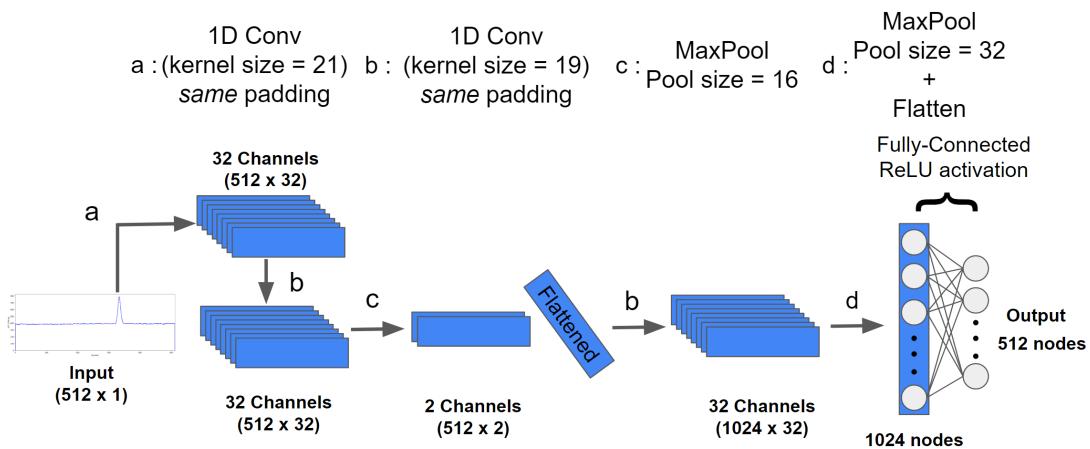


Figura 4.6: Arquitetura da rede neural que faz a inferência do fundo. O vetor de entrada deve ter dimensionalidade 512×1 . Todas as partes com convolução não possuem o parâmetro *bias*.

Tipicamente, CNNs consistem em um número de camadas convolucionais, eventualmente intercaladas com camadas *Max-Pooling*, e seguidas no final por um número de camadas *fully connected* (FC). Uma camada convolucional consiste de k filtros onde são computadas as operações de convolução entre o vetor de entrada e cada filtro da camada. A saída da camada convolucional consiste, então, em um conjunto de k sinais filtrados. O efeito resultante da aplicação dos filtros depende do tamanho do filtro e dos seus valores; um filtro pode, por exemplo, destacar algumas características do sinal de entrada. Logo, os resultados da saída das camadas convolucionais são conhecidos como mapas característicos (*feature maps*). Por exemplo, na rede neural de correção do sinal de fundo, cuja arquitetura está na figura 4.6, o sinal de entrada possui dimensão de 512×1 , o que significa um sinal com 512 *time buckets*, enquanto que a saída da camada convolucional possui dimensão de 512×32 , o que significa 32 sinais com 512 *time buckets* (32 mapas característicos, ou também canais). O número de filtros k é determinado empiricamente de forma à otimizar a performance da rede neural.

O tamanho dos filtros foram escolhidos entre 17 e 21 *time buckets* para cobrir totalmente a região com o pico. Importante notar que, pelo tamanho finito do sinal, não é possível aplicar a operação de convolução nas extremidades do sinal de entrada (primeiros e últimos pontos). Nas camadas convolucionais, se o sinal de saída precisa ser igual ao de entrada, então é escolhido o *same padding* (quando são adicionados zeros nas extremidades do sinal de entrada, de forma que a operação de convolução ocorra e o sinal resultante possua o mesmo tamanho do sinal de entrada).

Max Pooling é outra operação comum em CNNs. Seu efeito principal é de reduzir a dimensionalidade dos dados por meio de um processo baseado em uma janela com dado

valor de seleção (chamado de *pooling size*), selecionando sempre o maior dentre os valores possíveis de cada amostra. A seleção pode ser aplicada com respeito à dimensão dos *time buckets* dos sinais, assim como pode ser aplicada em respeito à dimensão dos canais. No nosso caso, o *pool size* escolhido foi 16 para reduzir o número de canais de 32 para 2. Isso reduz o número de parâmetros e consequentemente o tempo de treino. No final da rede neural, algumas camadas FC foram incluídas. Entre a última camada convolucional e a primeira camada FC foi aplicado uma planificação (ou *flat*) para diminuir a dimensionalidade da dimensão dos canais de 2 para 1. Na camada final, foi usada a função de ativação ReLU ($f(x) = \max(0, x)$) para garantir o valor mínimo em zero e para prevenir problemas com a minimização da função custo [59]. A função de ativação ReLU também é não linear que ajuda a rede neural a aprender operações não lineares, assim como no nosso caso.

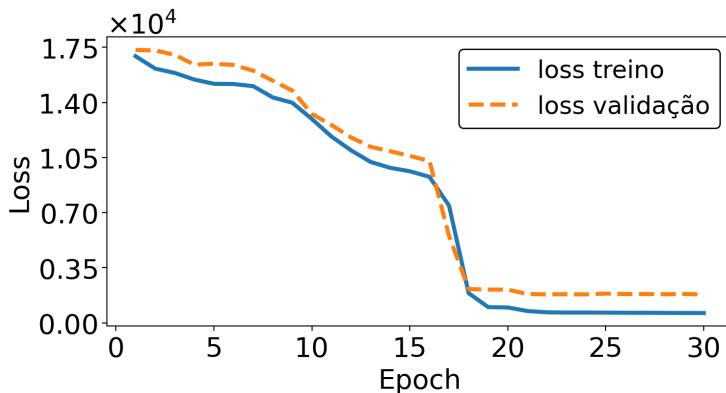
Para o treino foram usados 160.000 sinais pré analisados selecionados de forma aleatória. Um conjunto de dados menor de 40.000 sinais pré analisados também foram usados como validação para o modelo treinado. A CNN foi desenvolvida usando o TensorFlow 2 com mais de meio milhão de parâmetros treináveis. O treino foi realizado usando a GPU (graphics processing unit) NVIDIA Tesla P100 e durou cerca de 1 hora. O loss foi escolhido como sendo o erro quadrático médio, dado por

$$E = \frac{1}{N} \sum_{i=1}^N (x_i - \hat{x}_i)^2, \quad (4.5)$$

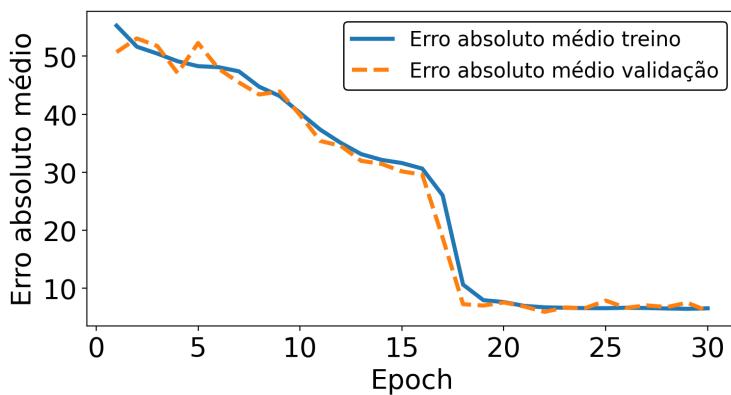
onde E é o erro quadrático médio, N é o número de pontos e x_i o ponto da saída da rede para ser comparado com o ponto original \hat{x}_i . O otimizador foi o ADAMAX [40], com learning rate de 0.0005 (parâmetro α da equação 3.6), e métrica para avaliação foi o erro médio absoluto, dado por

$$E = \frac{1}{N} \sum_{i=1}^N |x_i - \hat{x}_i|, \quad (4.6)$$

onde E é o erro absoluto médio, N é o número de pontos e x_i o ponto da saída da rede para ser comparado com o ponto original \hat{x}_i . Utilizamos 30 epochs (iterações de treino) e adotamos um batch size de 8. Os resultados do treino são mostrados na figura 4.7. Como podemos ver na figura, os treinos são bem validados a partir da epoch 20, quando começa um platô no *loss*.



(a) Loss dos dados de treino (linha contínua) e dos dados de validação (linha tracejada) em função da epoch no treino da rede dada pela figura 4.6.



(b) Erro absoluto médio dos dados de treino (linha contínua) e dos dados de validação (linha tracejada) em função da epoch no treino da rede dada pela figura 4.6.

Figura 4.7: Resultados do treino da rede neural representada na figura 4.6. A rede atingiu seu melhor resultado a partir da epoch 20.

A arquitetura da figura 4.6 (assim como as próximas desse capítulo) foi determinada de forma empírica. Uma arquitetura com menos passagens e/ou menos parâmetros fornece resultados menos adequados em comparação com a arquitetura apresentada. No caso de mais parâmetros e /ou passagens (consequentemente com aumento no tempo de execução), a rede neural não demonstrou melhora substancial.

Exemplos de resultados de previsões da rede neural estão na figura 4.8. A previsão do fundo possui um erro absoluto nos dados de treino de 6.5315 ADC Channels e nos dados de validação de 6.0783 ADC Channels. O sinal cru é subtraído do fundo, colo- cando o valor mínimo da subtração em 0. O erro médio absoluto de 200.000 sinais sem o respectivo fundo (resultante do algoritmo do TSpectrum) foi estimado em apenas 4.5 ADC Channels.

Rede neurais convolucionais têm a vantagem de usarem poucas variáveis e serem facilmente paralelizadas em sua execução [26]. Uma vantagem de redes neurais é o seu

tempo de execução. Empiricamente a rede neural pode processar 200.000 sinais em apenas 8 s (ou 25.000 sinais por segundo), sendo muito eficiente em tempo.

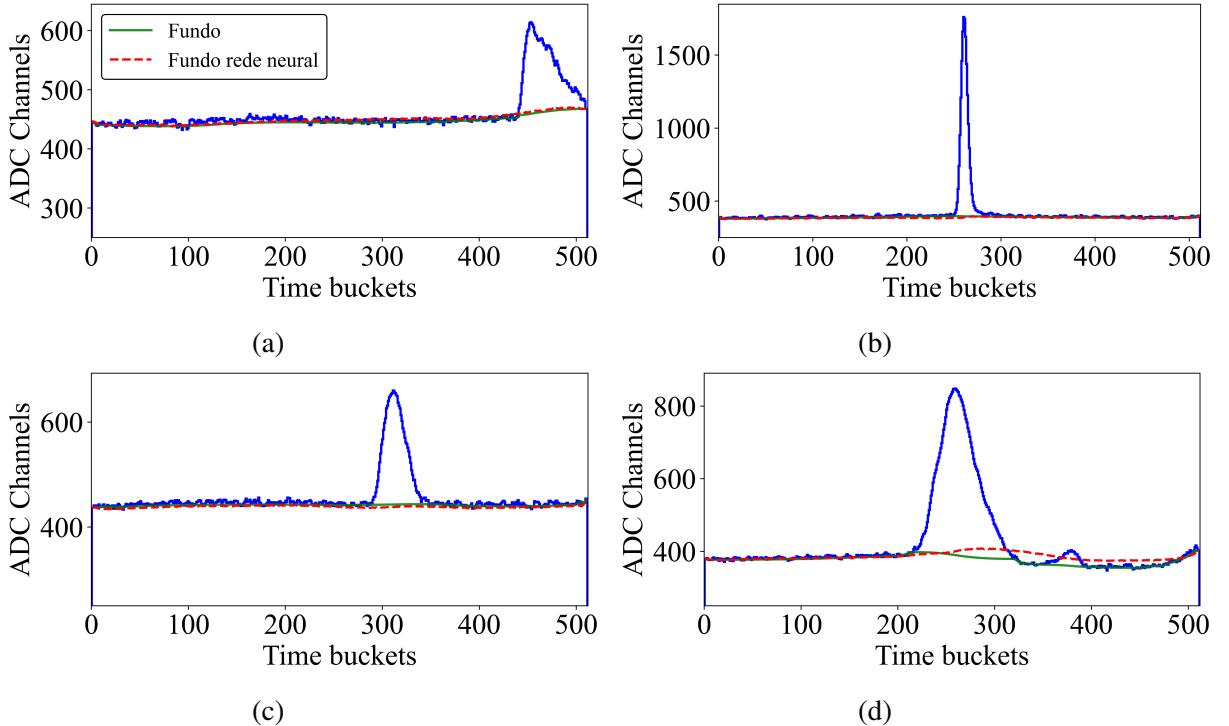


Figura 4.8: Exemplos da rede neural dada pela figura 4.6 em comparação com a saída do TSpectrum.

Nos exemplos mostrados nas figuras 4.8a, 4.8b e 4.8c os fundos dos sinais possuem grande flutuação e a rede neural se mostrou eficaz na previsão. No exemplo 4.8d a baseline do sinal é relativamente complexa pois o sinal do canal 300 a 500, varia em cerca de 50 unidades em y. Apesar da rede neural determinar o fundo, para esse caso, acima do fundo original, ao subtrair o espectro do fundo e colocar o valor mínimo em 0, o pulso presente entre os canais 200 e 300 é praticamente inalterado.

Com os resultados obtidos pela rede neural para a determinação do fundo, o próximo passo foi criar a rede neural que faz a deconvolução do espectro sem a baseline.

4.2.2 Rede neural para a deconvolução

Considerando a mesma abordagem da rede neural anterior criamos uma rede neural para realizar uma sequencia de convoluções e por fim uma camada *fully connected* com função de ativação ReLU, pois precisamos ter o valor mínimo do espectro em 0, assim como a saída do algoritmo de deconvolução do TSpectrum. Os filtros das convoluções precisam ter tamanho mínimo de duas vezes o sigma das gaussianas utilizadas para atu-

arem sobre cada pulso do espectro. A entrada da rede é o sinal já com o fundo subtraído e com mínimo em 0. A saída é o sinal após deconvolução dada pelo algoritmo GOLD DECONVOLUTION na biblioteca TSpectrum do ROOT, já mostrado na subseção 4.1.2. A figura 4.9 mostra a arquitetura da rede de deconvolução. A rede é a sequência de duas convoluções com 32 filtros, *valid padding* e filtros de tamanho 19 e 17, respectivamente. Sem o *valid padding*, nós temos a saída com tamanho reduzido, como o da segunda arquitetura mostrada na figura 4.9 (como o tamanho do filtro é 19, não é possível aplicar a convolução em 9 pontos em cada extremidade, então, o sinal resultante tem tamanho $494 = 512 - 18$). O *valid padding* se mostrou mais eficiente para a convergência da rede. Toda a rede foi construída usando o TensorFlow 2, possuindo 508.000 parâmetros treináveis [6].

Assim como na rede anterior, foram usados 160.000 sinais para treino e 40.000 para validação. O *loss* foi escolhido como sendo o erro quadrático médio, o otimizador foi o ADAM, com *learning rate* de 0.0005 porém com o parâmetro *clipnorm* igual a 0.45 que serve para alterar a norma do vetor gradiente da função do custo. Isso significa que, caso a norma do vetor do gradiente exceda 0.45, então o valor da norma é reajustado para o limiar (*threshold*) escolhido (0.45) [6]. Isso faz com que não ocorra problemas comuns como o gradiente sumir [59, 40], um dos problemas que teve que ser resolvido nessa rede. A métrica para avaliação foi o erro médio absoluto. Foram 75 *epochs* e o *batch-size* foi 8. Os resultados do treino estão na figura 4.10.

O treino foi realizado no Google Colaboratory [60] usando a GPU NVIDIA Tesla P100 e demorou cerca de 54 minutos. Os resultados do treino estão na figura 4.7, onde eles indicam que, visualmente, a rede neural consegue distinguir muito bem diferentes centróides presentes no pulso. Empiricamente, a rede é capaz de executar 200.000 sinais em 5.4 segundos (ou 37.000 sinais por segundo).

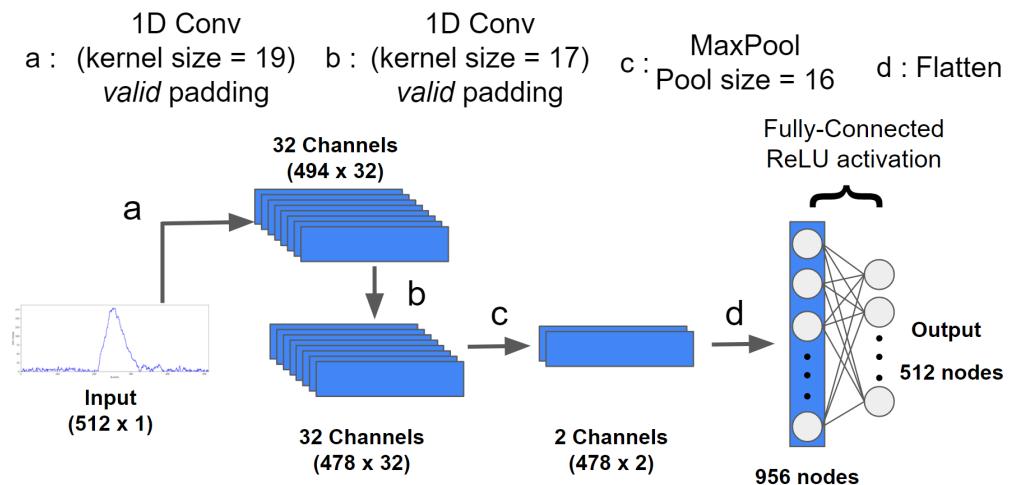


Figura 4.9: Arquitetura da rede neural que faz a inferência da deconvolução do espectro. O vetor de entrada deve ter dimensionalidade 512×1 . Todas as partes com convolução não possuem o parâmetro bias.

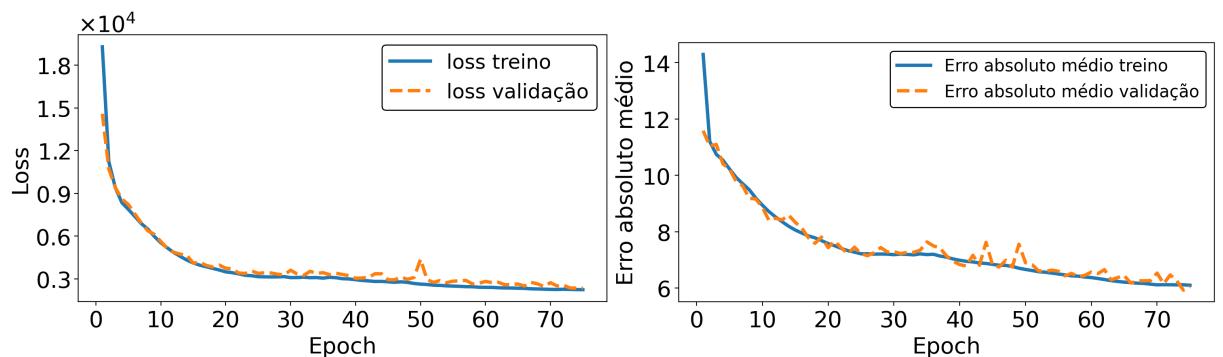


Figura 4.10: Resultados do treino da rede neural dada pela figura 4.6.

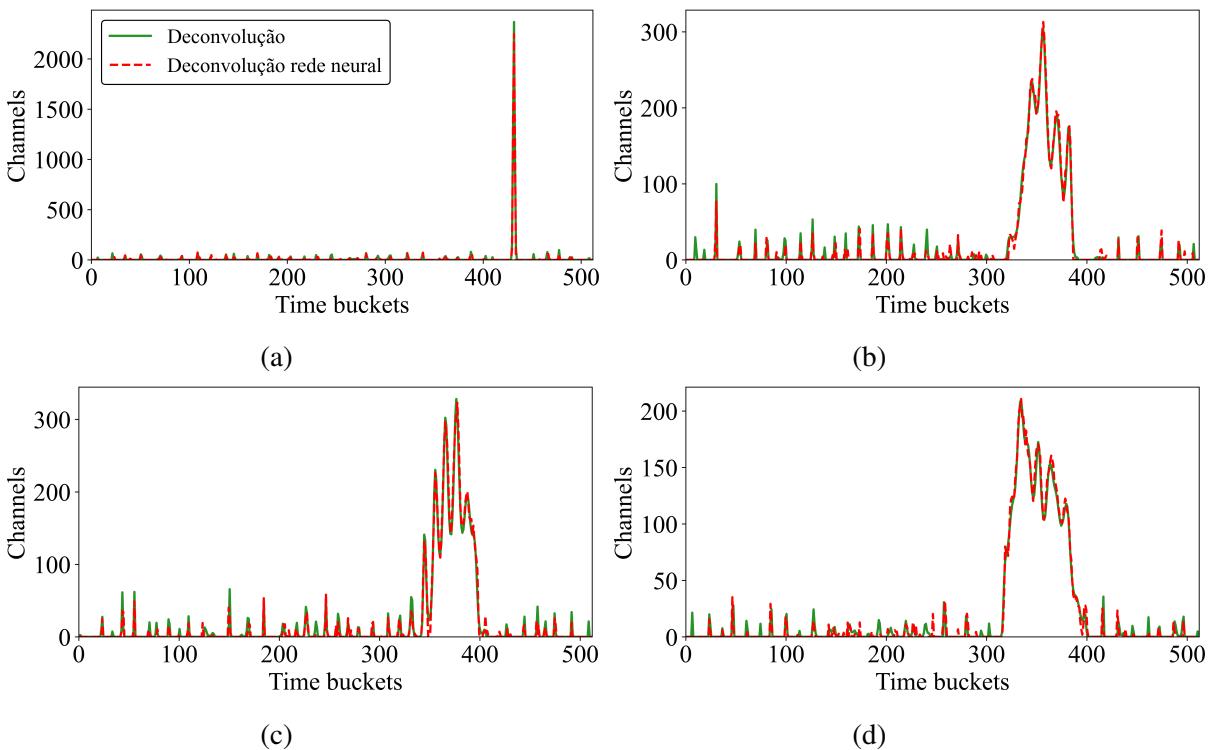


Figura 4.11: Exemplos de deconvolução da rede neural dada pela figura 4.6.

Com a rede neural para a deconvolução determinada e funcionando, a última etapa é a construção de uma rede neural para a identificação de picos, mostrada na 4.2.3.

4.2.3 Identificação de picos

A última etapa dessa análise com machine learning foi analisar e determinar uma solução para a identificação de picos. A identificação dos picos se tornou um desafio complexo, pois há uma variação muito grande na quantidade de picos por sinal e também há um desbalanço muito grande na quantidade de pontos comuns (aqueles que não são picos) e pontos que são picos. Por exemplo, caso haja um sinal que possui apenas um pico, deve-se identificar uma posição, gerar um valor de saída como 1. Por exemplo, para 512 pontos, um *time bucket* deve ter o valor 1 e os outros 511 terão o valor de saída como 0. Caso a rede determine que todos os pontos não são picos, ainda assim a acurácia binária seria maior que 99%. Isso é conhecido como desbalanço de classe [61].

Para corrigir esse desbalanço, foram acrescentados pontos simetricamente em torno do pulso, de forma que a somatória das amplitudes das regiões é equivalente a carga Q acumulada no ponto. Isso faz com que não seja preciso identificar um único ponto, mas sim uma região em torno do pico. Com isso podemos nos basear na ideia de segmentar o sinal para destacar regiões de interesse [62]. Segmentar significa ter uma rede neural com

a saída com o mesmo tamanho do vetor de entrada (512) e saída com valores entre 0 e 1, onde 1 indica uma região com um pico e 0 não. A figura 4.12 mostra um exemplo de um sinal após a deconvolução onde picos foram identificados com o algoritmo PEAK_FINDER e os pontos foram acrescentados simetricamente em torno dos picos para representar as regiões dos pulsos.

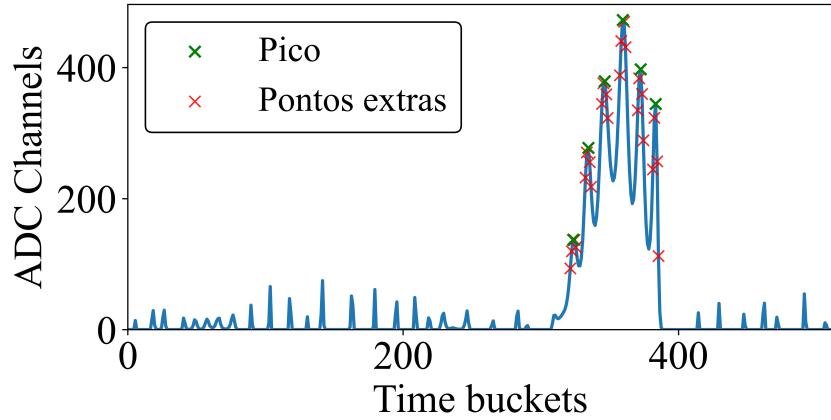


Figura 4.12: Sinal após a deconvolução que mostra o pico identificado mais os pontos adicionais que irão facilitar o trabalho da rede neural (evitar o desbalanço de classe). Foram acrescentados 2 pontos à esquerda e à direita.

A rede construída é a sequência de uma convolução com filtros de tamanho 13 e *same padding* seguida de *Max-Pooling* e uma camada *fully-connected*. Os neurônios na camada FC são seguidas pela função de ativação sigmoid, que limita os neurônios da camada de saída no intervalo [0, 1]. Assim, nesse caso nós trocamos a função custo *mean squared error* pela *binary cross entropy* [26]. Essa rede tem um total de 263.104 parâmetros treináveis. Para o treino foram usados sinais que possuíam entre 1 e 6 picos resultantes da saída do algoritmo PEAK_FINDER do SciPy, o que resultou em 120.024 dados para o treino e 30.006 para validação. A escolha pelos picos identificados pelo PEAK_FINDER ao invés do algoritmo de identificação do TSpectrum é devido a uma maior flexibilidade de ajuste fino do algoritmo, tornando a identificação de picos muito melhor[6]. O treino também foi realizado por uma GPU NVIDIA Tesla P100 e durou cerca de 8 minutos com 12 *epochs*. A arquitetura da rede está na figura 4.13 e os resultados do treino estão na figura 4.14.

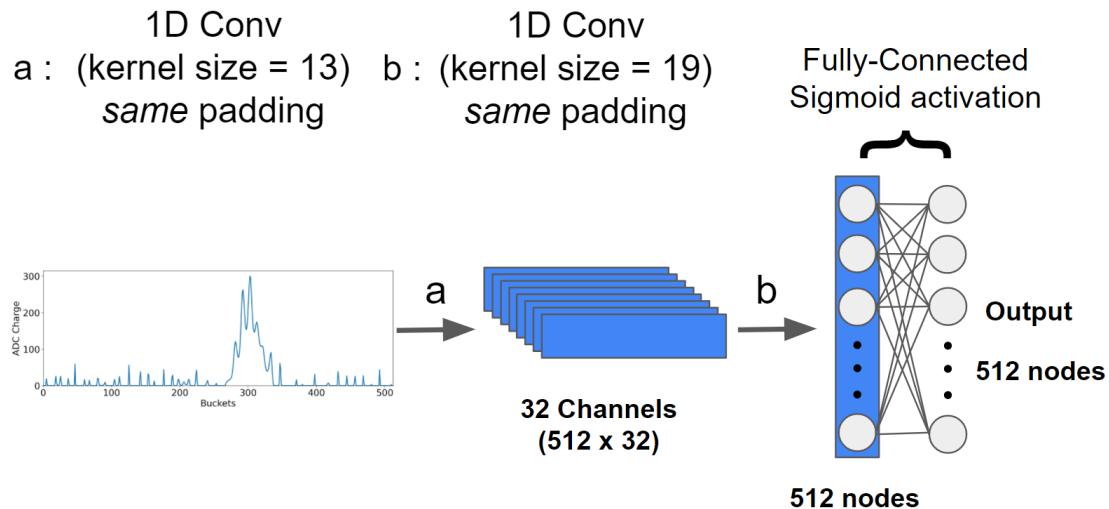


Figura 4.13: Arquitetura da rede neural que faz o recorte das regiões com picos. O vetor de entrada deve ter dimensionalidade 512×1 .

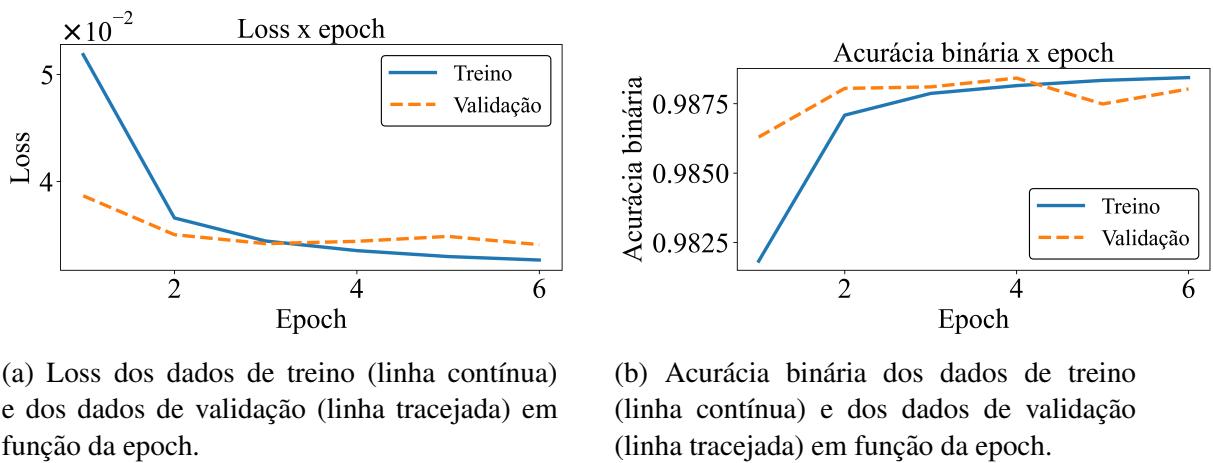


Figura 4.14: Resultados do treino da rede neural dada pela figura 4.13.

A saída da rede neural é um vetor de tamanho 512 com valores entre 0 e 1, onde valores maiores que 0.5 são considerados como pertencentes à um pico. Com as regiões identificadas podemos fazer uma média ponderada com o espectro de entrada para achar o centróide. O tempo de processamento da rede neural é de 150.030 sinais em cerca de 4.11 segundos (aproximadamente 36.500 sinais por segundo). Para determinar os picos a partir da saída da rede neural, o tempo é de aproximadamente 4.3 segundos, onde o algoritmo pode ser ainda mais rápido se for paralelizado. Resultados para picos identificados pela rede neural em comparação com o algoritmo PEAK_FINDER estão na figura 4.15.

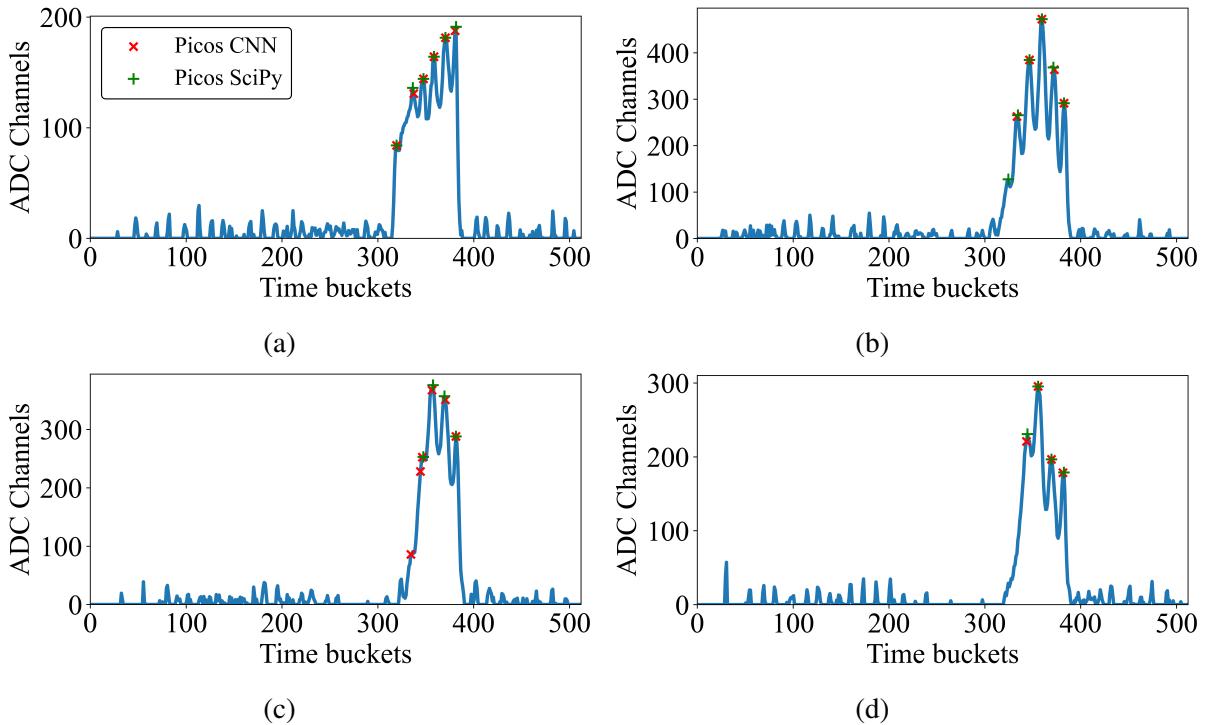


Figura 4.15: Exemplos de identificação de picos usando a rede neural, em comparação com a identificação feita pelo algoritmo presente no SciPy, mostrada na figura 4.13. Os centroides identificados pela rede neural estão em vermelho, e os centroides identificados pelo SciPy estão em verde. Em azul está o espectro sem fundo após a deconvolução, resultante

Para determinar carga acumulada Q_i de cada ponto associado à cada centroide i , foi usada a relação

$$Q_i = \sum_{i=2}^{i+2} f(t_i) \alpha_\sigma, \quad (4.7)$$

onde $f(t_i)$ é a amplitude do espectro no *time bucket* t da posição i e α_σ é uma constante real para calibrar o valor da área em função do sigma dos pulsos, que foi determinado empiricamente como $\alpha_\sigma = 1.2$.

Uma vez tendo determinado as três redes neurais, devemos ainda verificar o acoplamento das redes, a fim de analisar a qualidade dos algoritmos e os tempos de execução, para comparar com os algoritmos discutidos nas subseções 4.1.1 e 4.1.2.

4.2.4 Acoplando as redes neurais

Usando novamente o TensorFlow 2 pode-se carregar as redes neurais discutidas nas subseções 4.2.1, 4.2.2 e 4.2.3, já treinadas e usar como se fosse uma única rede neural.

Acoplando as três arquiteturas, temos a nova arquitetura mostrada na figura 4.16. O *input* da rede é o sinal cru sendo que o *output* da rede neural é o um vetor de tamanho 1024, que possui a segmentação e o sinal após a deconvolução, pois as duas informações são necessárias para determinar os centróides.

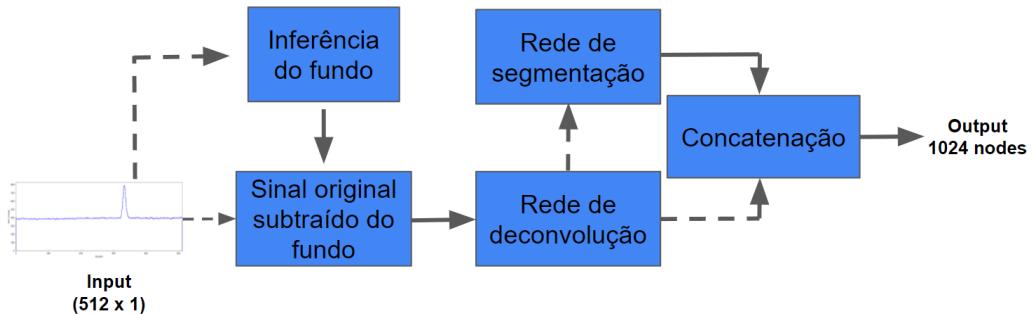


Figura 4.16: Arquitetura da rede neural que faz a inferência da baseline, em seguida faz a deconvolução do espectro sem o fundo e por fim faz a segmentação do sinal. O resultado da segmentação e da deconvolução são concatenados na parte final da rede neural. O vetor de entrada deve ter dimensionalidade 512 x 1.

A rede final é apenas a sequencia das redes anteriores, ou seja, o espectro passa pelo cálculo da baseline, então o espectro original é subtraído dessa *baseline* (colocando o valor mínimo em 0) para passar pela etapa da deconvolução e em seguida o sinal é segmentado. Na etapa final, o resultado da segmentação e da deconvolução são concatenados em um único vetor, para assim determinar os centroides. Pelo fato de cada parte ser treinada de modo separado não há necessidade de treinar a rede unificada, apenas carregar as variáveis das redes neurais treinadas de cada parte. A rede unificada possui 1.316.640 de parâmetros.

A saída da rede acoplada para o espectro após a deconvolução foi comparada com a saída de referência, que é o espectro após a deconvolução mostrado na subseção 4.1.2. Para uma análise comparativa quantitativa usamos os valores do erro médio absoluto, cujo valor médio para os 200.000 sinais, foi de 7.45 ADC Channels. Com relação à incerteza, ela foi estimada como o desvio padrão da diferença de cada time bucket do sinal tido como referência e o sinal após a rede neural acoplada. A incerteza é da ordem de 6% da amplitude do sinal no ponto.

Com relação a eficiência em tempo, a rede neural da figura 4.16 processa 200.000 sinais em 12 segundos, usando a GPU NVIDIA Tesla P100. Somando esse tempo com a determinação dos centroides, que é de aproximadamente 4.3 segundos, o tempo total para processar 200 mil sinais é de cerca de 16.3 segundos, cerca de 90 vezes mais rápido que os métodos mostrados na seção 4.1. O resultado final dessa rede neural são a geração de pontos no espaço definidos pelos parâmetros, x , y e z . Um evento, correspondente à uma

reação nuclear no alvo ativo gera vários pontos formando o que chamamos de nuvens de pontos. Essas nuvens de pontos definem então a trajetória das partículas produzidas por uma reação nuclear. Exemplos da reconstrução das nuvens de pontos usando a rede neural da figura 4.16 são apresentados na figura 4.17d.

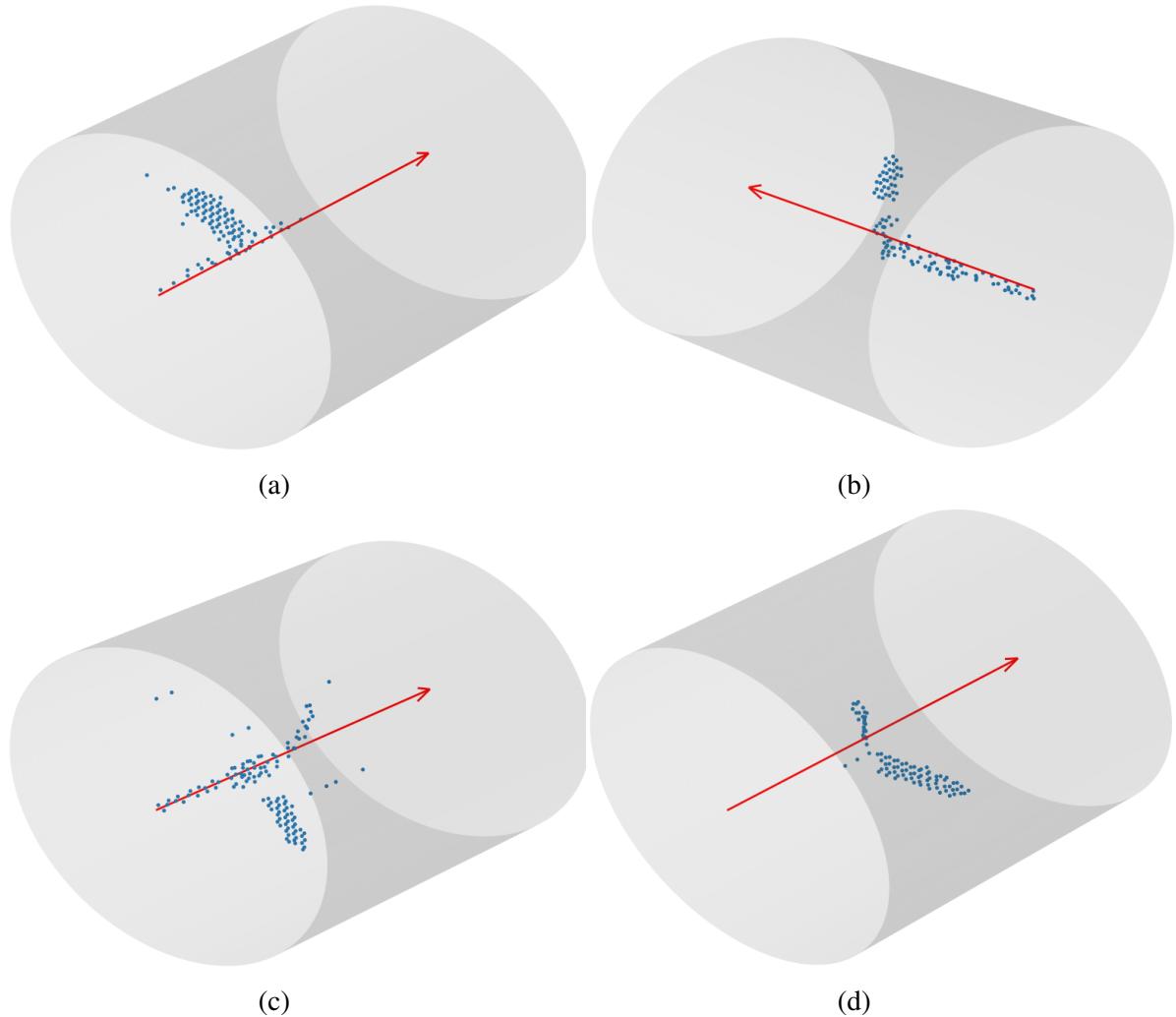


Figura 4.17: Exemplos de eventos reconstruídos através da análise dos sinais com machine learning. A seta vermelha indica o sentido do feixe. Diferentes nuvens de pontos (eventos) correspondem a diferentes trajetórias de partículas sendo produzidas por diferentes reações nucleares.

Com as redes neurais desenvolvidas, o consumo de tempo para o processamento dos pulsos diminuiu em cerca de 90 vezes em relação à algoritmos com programação tradicional mostrados na seção 4.1, abrindo possibilidades para a análise em tempo real de um experimento com alvo ativo usando grande granularidade (milhares de canais). Com as nuvens de pontos reconstruídas, pode-se extrair propriedades físicas dos eventos, processo que está descrito no capítulo 5 a seguir.

Capítulo 5

Análise das nuvens de pontos

As nuvens de pontos representam eventos de reações nucleares dentro do alvo ativo. O objetivo desta etapa, do ponto de vista computacional, foi de identificar *clusters* (aglomerados) tridimensionais de pontos, que no caso do pAT-TPC (sem a presença de campo magnético) devem seguir um comportamento de trajetórias retilíneas. Já do ponto de vista físico, o objetivo é extrair as informações das reações nucleares (energia E , momento \vec{p} e comprimento L) a partir das trajetórias das partículas. Para tanto, foi necessário distinguir cada trajetória identificada e determinar o vértice da reação. Após a identificação de cada trajetória, foram selecionados os eventos correspondentes ao canal de reação do breakup cujas partículas seguiam uma trajetória específica.

Assim como no capítulo 4, esse capítulo descreve o uso de algoritmos de machine learning para a análise, determinação e identificação das trajetórias a partir das nuvens de pontos. Os algoritmos usados foram tanto não supervisionados (algoritmos de *clustering*) e supervisionados (redes neurais), com diferentes objetivos.

Na seção 5.1 está mostrado o processo de identificação de trajetórias (*tracking*) com algoritmo de machine learning não supervisionados. Na seção 5.2.1 estão mostradas alternativas com outros algoritmos de machine learning, que não foram utilizadas para a análise dos dados, mas que abrem um grande leque de possibilidades para estudos futuros.

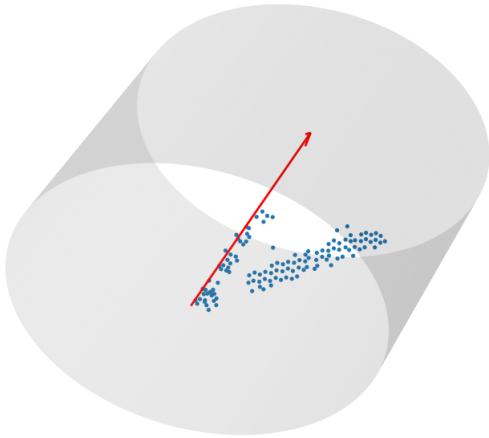
5.1 Identificação de trajetórias

Uma vez reconstruídas as nuvens de pontos de cada evento, é necessário classificar cada uma das trajetórias (*tracks*) das partículas dentro do gás. A figura 5.1a mostra um exemplo de uma nuvem de pontos. A classificação das trajetórias pode ser realizada a partir de algoritmos de machine learning não supervisionados. Nessa figura é notável ao

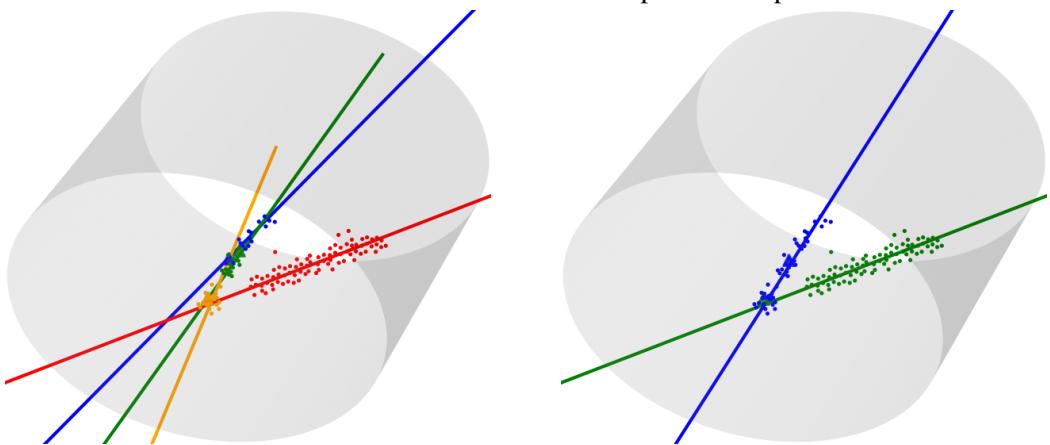
olho humano que os pontos formam estruturas de aglomerados. Pode-se associar uma reta a essas estruturas que corresponderiam a trajetória das partículas dentro do alvo gasoso. Essas retas foram identificadas com algoritmos de *clustering*. Devemos ressaltar que as nuvens de pontos não tem um tamanho ou número de pontos definidos. O número de pontos pode variar de 30 até cerca de 200, não considerando pontos *outliers* (não pertencem a nenhum aglomerado).

Para melhor definirmos os aglomerados devemos primeiramente eliminar ou diminuir os *outliers*. Para tanto, foram usados dois filtros. O primeiro filtro exclui pontos baseado em sua carga, colocando um limiar onde pontos com carga $Q < 110$ são descartados. O segundo filtro, chamado de OUTLIER REMOVAL, elimina pontos considerados *outliers* globais (que não fazem parte de nenhum aglomerado), de modo que, caso um ponto não possua um número mínimo de vizinhos $n_{or} = 4$ em um raio de distância tridimensional $d_{or} = 12$ mm, então ele é descartado. O OUTLIER REMOVAL está presente na biblioteca Open3D [63] no Python e funciona excluindo pontos muito isolados uns dos outros.

Existem diversas opções de algoritmos de identificação de cluster disponíveis, como o *Density-based spatial clustering of applications with noise* (DBSCAN) [64], porém nossa escolha foi baseada na performance em tempo. Portanto, o algoritmo usado foi o *Hierarchical DBSCAN* (HDBSCAN) [65, 66]. O algoritmo tem como entrada a nuvem de pontos e parâmetros esperados para a densidade dos aglomerados presentes no conjunto de pontos, como a densidade mínima e máxima de pontos nos aglomerados. O algoritmo retorna vetores que tornam possível a identificação dos diferentes aglomerados. Os aglomerados passam por um ajuste por mínimos quadrados para determinar o versor e um ponto arbitrário que determinam a reta tridimensional relacionada com a trajetória. A figura 5.1b mostra o resultado da aplicação do HDBSCAN na nuvem de pontos mostrada na figura 5.1a. As nuvens de pontos reconstruídas não possuem, em certos casos, densidade de pontos o suficiente para resultar em 100% de acurácia de identificação correta de todas as trajetórias do evento.



(a) Exemplo de evento analisado. Os pontos em azul são das partículas detectadas pelo TPC, a seta vermelha indicando o sentido do feixe e o TPC está representado pelo cilindro cinza.



(b) Evento com as identificações sem a correção. As três retas de cores amarela, verde e azul são de um único cluster.

(c) Evento corrigido. Agora o evento possui as duas retas corretas, a azul e a verde.

Figura 5.1: Sequência de análise de um evento. Em 5.1a temos o evento que é recebido para ser analisado, em 5.1b temos o mesmo evento após o HDBSCAN (antes da correção) e 5.1c mostra depois da correção. As cores das retas são arbitrárias e servem apenas para a diferenciação.

O HDBSCAN pode apresentar falhas na análise dos aglomerados e identificação das trajetórias, como visto na figura 5.1b, em que, por exemplo, em que uma nuvem de pontos acaba sendo identificada e dividida em três clusters muito próximos. Para esses casos, foi aplicada uma correção da saída do *clustering*. A correção foi feita comparando em pares todos os aglomerados resultantes do algoritmo e unificando os subconjuntos muito semelhantes [14]. Do ponto de vista computacional, esse problema é abordado avaliando a semelhança entre dois aglomerados usando métricas, como a distância de Jaccard [67] e o coeficiente de silhueta [68]. A correção feita se dá em duas etapas: primeiro comparando os versores entre duas retas e depois verificando se a condição da equação 5.1 é satisfeita. Caso a diferença absoluta entre os ângulos com relação ao versor $(0, 0, 1)$ (direção do

feixe de ^{17}F) seja menor que 9° (determinado empiricamente), então as duas retas serão combinadas se obedecerem a condição dada por

$$\sum_{i=0}^{N_1} \frac{d_{i2}}{N_1} < \alpha d_{min}, \quad (5.1)$$

onde N_1 é o número de pontos da reta 1, d_{i2} a distância do ponto i da reta 1 em relação à reta 2, α é um parâmetro com valor a ser escolhido e d_{min} é a distância mínima do ponto a reta. Os valores foram determinados empiricamente, tais que $\alpha = 1.75$ e $d_{min} = 15$ mm. A figura 5.1c mostra o resultado da correção baseada nesses critérios no resultado anterior mostrado na figura 5.1b.

Após a correção, é necessário classificar cada reta como sendo feixe, ou uma partícula originada de uma reação nuclear. O feixe incide na câmara possui um ângulo muito pequeno com relação ao versor $(0, 0, 1)$, que é o versor de incidência do feixe. Além disso, mesmo que o ângulo seja pequeno, a reta do feixe cruza o plano da janela do TPC próximo ao ponto mais provável da entrada o feixe. Por sua vez, o ponto mais provável foi determinado usando a posição média da projeção dos pontos de um conjunto de eventos no plano x - y . A posição inicial mais provável obtida para o feixe foi $x_f = -3.4$ (6.7) mm e $y_f = -0.9$ (6.3) mm. A incerteza é alta devido às dimensões do feixe (aproximadamente 5 mm). Portanto, se o ângulo entre o versor \hat{v}_i de uma reta r_i for menor que 5° (determinado de modo empírico novamente) e a distância d entre o ponto P_i que intercepta o plano e o ponto $(x_f, y_f, 0)$ for menor que 15 mm (pouco mais que duas vezes a incerteza de cada ponto), então a reta foi considerada como o feixe do evento. No caso de não satisfazer essas condições, então ela foi classificada como uma possível partícula originada da reação do feixe com o gás.

É importante ressaltar que alguns eventos são caracterizados por apenas uma trajetória reta não sendo possível determinar o projétil ou o ejétil. Um exemplo desse tipo de evento é mostrado na figura 5.2. Nesse caso, foi necessário assumir as propriedades da reta mais provável para o feixe, ou seja, uma reta que passe pelo ponto $(x_f, y_f, 0)$ e que tenha um versor $(0, 0, 1)$.

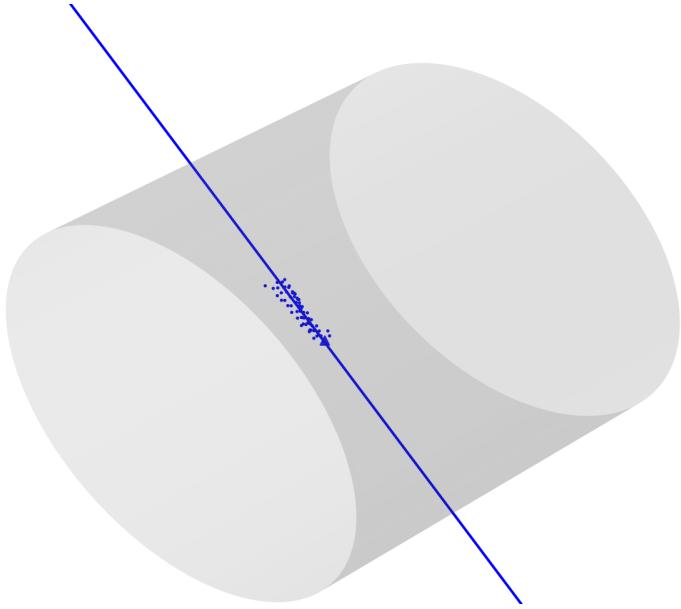


Figura 5.2: Evento em que não foi identificado o feixe, apenas a partícula espalhada. O triângulo azul é o local calculado do vértice de reação dado pela equação 5.6.

Para podermos selecionar um determinado tipo de reação precisamos primeiramente caracterizá-la. Uma reação nuclear é caracterizada pela incidência de partículas do feixe (em movimento), partículas do alvo (em repouso) e as partículas emergentes da reação. As partículas da reação são emitidas considerando uma determinada cinemática. Para determinarmos completamente a cinemática do evento foi necessário determinar o vértice da reação (ponto em que a reação ocorre) que corresponde ao ponto de cruzamento entre a trajetória da partícula e a direção do feixe. Portanto, o vértice de reação é o ponto médio do menor segmento de reta que conecta a trajetória da partícula espalhada ou emergente da reação e o feixe, no ponto de menor distância entre as retas. O vértice é de grande importância para que possamos determinar o tipo de reação nuclear que ocorreu. Para deduzir uma expressão que possamos usar para determinar o vértice da reação temos que considerar as seguintes equações das retas \vec{P}_1 e \vec{P}_2 , que podem ser retas quaisquer (que serão atribuídas às partículas), como vetores:

$$\begin{aligned}\vec{P}_1 &= \vec{A}_1 + \vec{V}_1 * t_1 \\ \vec{P}_2 &= \vec{A}_2 + \vec{V}_2 * t_2,\end{aligned}\tag{5.2}$$

onde \vec{A}_1 e \vec{A}_2 são pontos arbitrários que pertencem as retas 1 e 2, respectivamente, \vec{V}_1 e \vec{V}_2 são os versores, t_1 e t_2 são os hiperparâmetros escalares das retas.

Como o vértice de reação é o ponto médio do menor segmento de reta que conecta

as duas retas, então é preciso determinar o versor \vec{V}_c desse segmento de reta. Então, \vec{V}_c é dado por

$$\vec{V}_c = \frac{\vec{V}_1 \times \vec{V}_2}{|\vec{V}_1 \times \vec{V}_2|}. \quad (5.3)$$

Podemos então construir a reta \vec{P}_3 que possui as características do menor segmento de reta que conecta \vec{P}_1 e \vec{P}_2 . Essa reta deve começar no ponto de menor distância da reta 1 e terminar no ponto de menor distância da reta 2. Ou seja, tem-se o seguinte sistema linear:

$$\vec{A}_2 + \vec{V}_2 * \tilde{t}_2 = \vec{A}_1 + \vec{V}_1 * \tilde{t}_1 + \vec{V}_c * \tilde{t}_3.$$

Rearranjando temos que

$$\vec{V}_1 * \tilde{t}_1 - \vec{V}_2 * \tilde{t}_2 + \vec{V}_c * \tilde{t}_3 = \vec{A}_2 - \vec{A}_1, \quad (5.4)$$

onde \tilde{t}_1 , \tilde{t}_2 e \tilde{t}_3 são os hiperparâmetros a serem determinados. Caso \vec{V}_1 seja paralelo à \vec{V}_2 , então não há solução (não há vértice de reação, ou seja, não há uma reação nuclear em comum entre a partícula e o feixe analisado). Com a solução do sistema pode-se obter os pontos de menor distância nas duas retas:

$$\begin{aligned} \vec{P}_1 &= \vec{A}_1 + \vec{V}_1 * \tilde{t}_1 \\ \vec{P}_2 &= \vec{A}_2 + \vec{V}_2 * \tilde{t}_2. \end{aligned} \quad (5.5)$$

Com isso, é possível determinar que o vértice de reação \vec{V}_r é dado por

$$\vec{V}_r = \frac{1}{2}(\vec{P}_1 + \vec{P}_2). \quad (5.6)$$

Pode-se definir a distância de máxima aproximação d_{max} das retas, dada por:

$$d_{max} = |\vec{P}_1 - \vec{P}_2|. \quad (5.7)$$

Da equação 5.7 foi estabelecido um limite superior para d_{max} tal que valores menores que esse limite indicam uma reação nuclear em comum entre a partícula e o feixe. O valor foi determinado empiricamente e foi definido como $d_{max}^{sup} = 25$ mm. Trajetórias cuja distância máxima de aproximação excedia d_{max}^{sup} , então a trajetória é descartada. Além disso, é necessário garantir que a reação ocorreu dentro da área efetiva da câmara e para tanto impomos os limites: $|x| < 140$ mm, $|y| < 140$ mm e $|t| < 512$.

Para completar essa etapa, foi necessário determinar a carga total Q_T e o comprimento L da trajetória. A carga total Q_T é definida como

$$Q_T = \sum_{i=1}^N Q_i, \quad (5.8)$$

onde Q_i é a carga do i -ésimo ponto de um conjunto com N pontos. O comprimento é definido como

$$L = \max \left\{ d_{i,\vec{V}_r} \right\}, \quad (5.9)$$

onde d_{i,\vec{V}_r} é distância entre o ponto i pertencente a trajetória até o vértice de reação \vec{V}_r .

Essa não foi a única abordagem utilizada nesse trabalho para a determinação do vértice e portanto para a determinação da reação nuclear. Na seção 5.2 mostramos alternativas para se buscar, com as nuvens de pontos crua, eventos específicos que ocorreram no experimento, sem ter que previamente buscar trajetórias e também outros usos de machine learning para a análise.

5.2 Abordagens alternativas

O objetivo da análise das nuvens de pontos foi de selecionar eventos que possuíam três trajetórias, uma sendo o feixe, e as outras duas de partículas que surgiram da reação nuclear do feixe com o gás.

5.2.1 Identificação de eventos com machine learning

Para evitar o processo de busca de aglomerados nas nuvens de pontos que não possuem eventos de interesse (como o breakup), é possível usar redes neurais supervisionadas capazes de processar nuvens de pontos tridimensionais. A rede neural usada para esse processo chama-se POINTNET [69]. A POINTNET é uma rede neural desenvolvida para classificar e segmentar imagens em 3D utilizando nuvens de pontos que representam uma forma 3D do objeto. A arquitetura dessa rede neural está na figura 5.3.

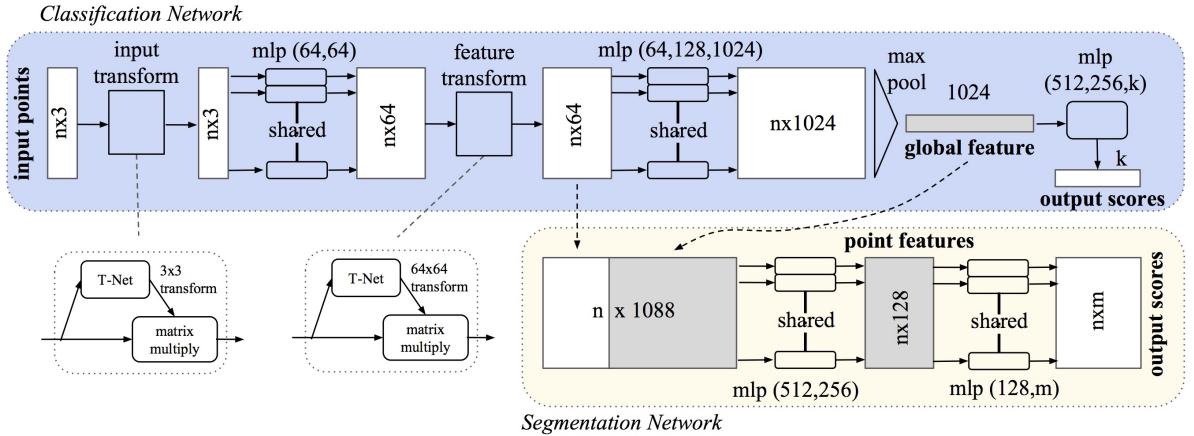


Figura 5.3: Arquitetura da POINTNET. A rede de classificação tem os dados de entrada com n pontos com 3 coordenadas, onde são aplicadas sequências de transformação que são agregadas por uma camada de *max pooling*. A saída é a classificação para k classes possíveis. A rede de segmentação é uma extensão da rede de classificação, classificando ponto a ponto a nuvem de pontos, em m classes possíveis. Mais detalhes sobre a arquitetura podem ser encontrados na Ref. [69]

A POINTNET é uma rede neural capaz de processar nuvens de pontos, entendendo que a estrutura dos dados é invariante à troca de pontos e invariante sobre transformações (rotação e translação) [70], pode ser usada tanto para classificação quanto segmentação semântica de nuvens de pontos [69]. A classificação é realizada para a nuvem de pontos como um todo, já a segmentação semântica é a classificação ponto a ponto da nuvem de pontos [69]. Para o uso da POINTNET para classificação, foi criado um banco de dados para o treino da rede neural que tem como entrada a nuvem de pontos e como saída o número de trajetórias presentes no evento. Um dos pontos negativos da POINTNET é a necessidade de que os dados de entrada tenham um tamanho fixo [69]. No caso dos dados desse trabalho, o número de pontos por evento não é fixo. Para resolver isso, impomos para os nossos dados um tamanho fixo $n = 300$ pontos para todas as nuvens de pontos. Eventos com menos pontos eram preenchidos com pontos repetidos da mesma nuvem para completar até o tamanho n . Eventos com menos do que 100 pontos e mais que 300 pontos foram descartados pois geralmente são ruídos e a saturação do detector devido à faíscas.

Para determinar o número de trajetórias nos eventos para o treino da rede neural, foi utilizado o estimador robusto (que estima quais as trajetórias presentes nas nuvens de pontos) que chamei de PROTOTYPE-RANSAC (RANdom SAmple Consensus), que é uma variação do RANSAC [71, 14]. A escolha dele no lugar do HDBSCAN é pela melhor acurácia na identificação de trajetórias, apesar de ser cerca de 4 vezes mais lento.

As nuvens de pontos (sem o acréscimo de pontos para a POINTNET) ainda passam pelos dois filtros e critérios de correção mostrados na seção 5.1. O algoritmo 1, localizado no apêndice B, mostra o funcionamento do P-RANSAC. O algoritmo seleciona dois pontos de modo aleatório e determina o versor \hat{v} e o ponto P_b que descrevem a única reta r que passa pelos dois pontos. A reta é selecionada caso tenha um número mínimo de pontos $N_{min} = 24$ (chamado de *inliers*) e tenha o mínimo (com relação aos outros conjunto de pontos) da estimativa C dada por [14]

$$C = \sum_{i=0}^N \frac{d_i^2}{N}, \quad (5.10)$$

onde N é o número total de pontos de uma reta e d_i é a distância do i-ésimo ponto à reta. O número de iterações do algoritmo foi determinado empiricamente e o melhor valor foi de 700.

Para a saída foram escolhidos eventos que possuem de 0 até 5 trajetórias. Com os dados para a camada de saída construídos, a rede para classificação foi treinada, onde a função custo foi a *categorical cross entropy* [39] e o otimizador o ADAM [40]. A acurácia de 70% quando concluído o treino. Exemplo do resultado da aplicação da rede neural em nuvens de pontos está na figura 5.4.

Trajetórias = 3. Resultado PointNet = 3

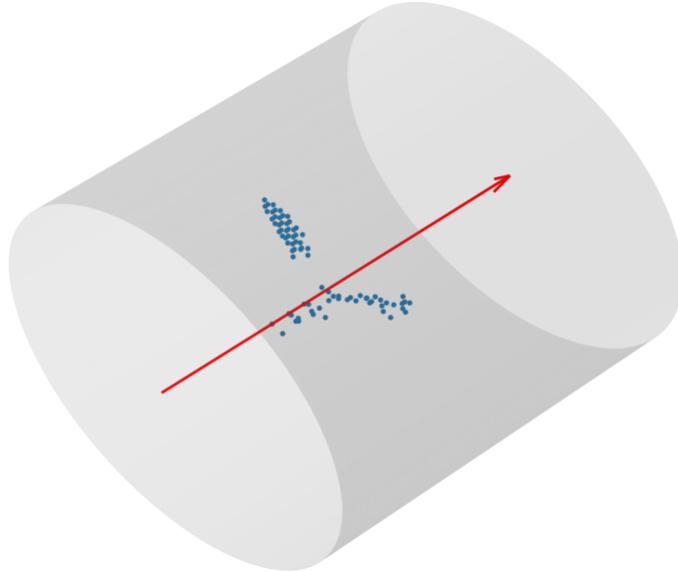


Figura 5.4: Nuvem de pontos que possui 3 trajetórias para serem identificadas e a rede neural de classificação calculou que haviam 3 trajetórias, o que indica um resultado correto do algoritmo. A seta vermelha indica a direção e o sentido do feixe.

Eventos com 4 ou 5 trajetórias ocorrem com uma frequência muito menor (menos

que 5% do total de dados) que eventos de 1 e 2 trajetórias (cerca de 85% do total de dados), caracterizando o banco de dados com o problema de desbalanço de classe (já discutido na subseção 4.2.3), o que pode ter prejudicado o treino para identificar eventos mais incomuns.

5.2.2 Identificação de *outliers*

Um possível uso para a rede de segmentação semântica é para classificar pontos como *inliers* ou *outliers*, semelhante ao uso do algoritmo OUTLIER REMOVAL da biblioteca Open3D [63]. A diferença é que com a POINTNET pode-se incluir *outliers* locais (que fazem parte de outro aglomerado) [70], não apenas os globais para serem identificados.

Para a saída da rede neural, os pontos classificados como *outliers* globais ou locais possuem valor 0 e pontos que são *inliers* (pertencem à alguma trajetória) possuem valor 1. A rede de segmentação foi treinada, com a função custo sendo a *binary cross entropy* e a métrica a acurácia binária. A acurácia da rede foi de aproximadamente 93%, se mostrando uma boa alternativa para eliminação de ruído nos eventos. A figura 5.5 mostra o resultado da aplicação da rede de segmentação em uma nuvem de pontos.

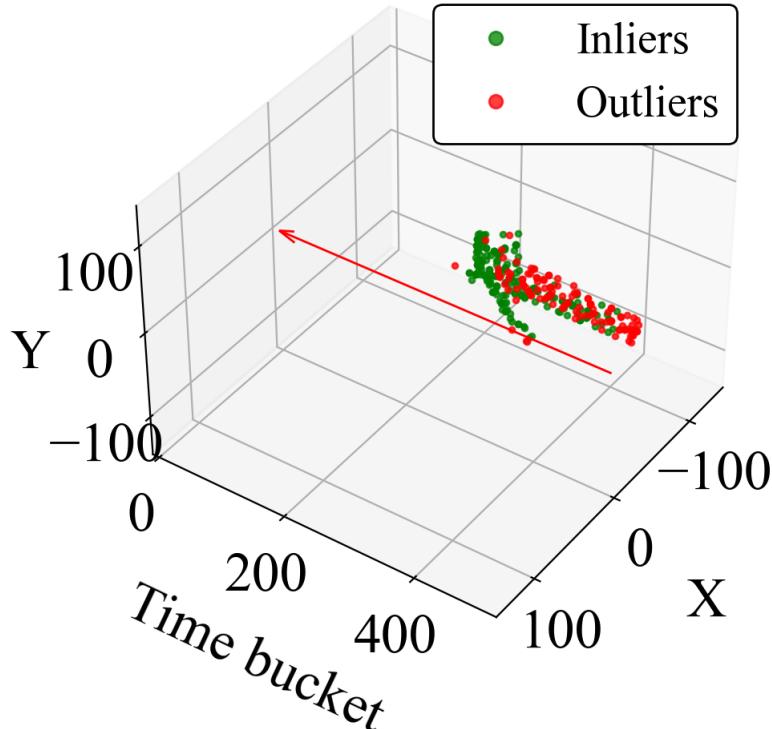


Figura 5.5: Identificação de *outliers* com a rede neural de segmentação. A rede neural foi capaz de identificar os *outliers* desse evento com 95% de acurácia.

Existem outras redes neurais que são capazes de lidar com nuvens de pontos, como a

POINTNET++ [72] e DYNAMIC GRAPH CNN [73]. Para a abordagem de identificação de trajetórias de forma direta, uma opção é utilizar a rede neural CONTRASTNET [74], pois a rede é capaz de fazer *clustering* nos dados, retornando os diferentes aglomerados existentes nos dados. Num futuro, estamos pensando em utilizar esses tipos de ferramentas para realizar *clustering* em nuvens de pontos em outros experimentos usando múltiplos canais de reação. No entanto, esse projeto está além dos objetivos desta dissertação.

A classificação e ajuste de trajetórias apresentados no presente capítulo permitem obter informação da cinemática das reações nucleares e extrair as respetivas distribuições angulares. Essa análise é discutida no próximo capítulo.

Capítulo 6

Resultados

Uma vez tendo determinado as trajetórias das partículas e classificação das possíveis reações podemos selecionar a reação de interesse. Nesse nosso trabalho estamos interessados na reação de breakup, que corresponde a quebra do feixe de ^{17}F em $^{16}\text{O} + p$ devido a interação com o alvo de ^4He . Para que possamos estudar a reação de breakup de interesse precisamos obter a distribuição angular, ou seja, a probabilidade de que a reação aconteça em função do ângulo. Nesse capítulo vamos descrever como podemos construir as distribuições angulares da reação de breakup do ^{17}F a partir da análise das nuvens de pontos, como foi apresentado no capítulo 5.

6.1 A cinemática da reação

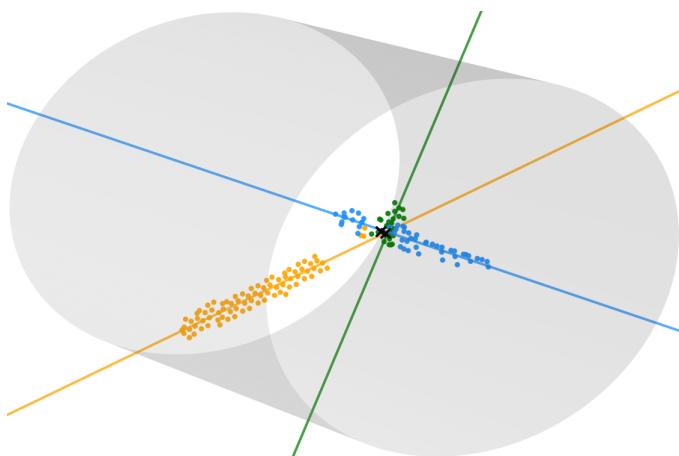
O processo de seleção e classificação de reações nucleares no alvo ativo é complexo e envolve várias etapas como: identificação de trajetórias e multiplicidade, determinação do vértice de reação, identificação de partículas e reconstrução da energia e ângulos da reação. As subseções a seguir descrevem essas etapas.

6.1.1 Reconstrução do vértice da reação

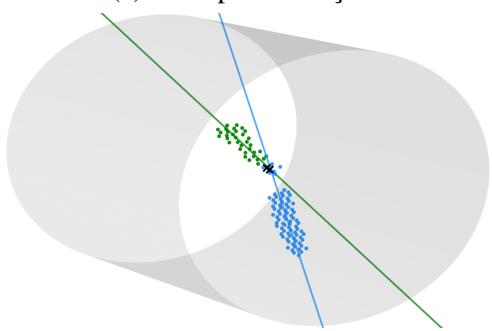
Para identificar e selecionar as reações nucleares que ocorrem dentro do alvo ativo é preciso identificar as trajetórias das partículas dentro do gás. O processo de determinação dessas trajetórias foi descrito na seção anterior, bem como a descrição de como obter o vértice da reação (que é o ponto em que efetivamente ocorre a reação nuclear). Isso é feito com estimadores robustos, como o algoritmo P-RANSAC (algoritmo 1, apresentado no apêndice B e também na referência [14]), que é capaz de identificar os diferentes aglomerados mesmo na presença de *outliers*.

Após a identificação e ajuste das trajetórias, é preciso determinar o vértice da reação para cada evento. O vértice é o ponto no espaço no qual ocorre a reação nuclear e pode ser determinado pela extrapolação das trajetórias na região central do alvo ativo. O ponto de interseção (ou mais próximo) entre duas retas é calculado como mostra a equação 5.6. O feixe incide na parte central pelo eixo axial do alvo, região cujo ganho do micromegas é baixo para evitar a saturação dos sinais (devido à passagem do feixe). Portanto, para calcular o vértice de reação, é necessário extrapolar as trajetórias na região central. A partir do vértice, então, é possível calcular o comprimento L da trajetória (dado pela equação 5.9), mesmo com a ausência de informação na região central.

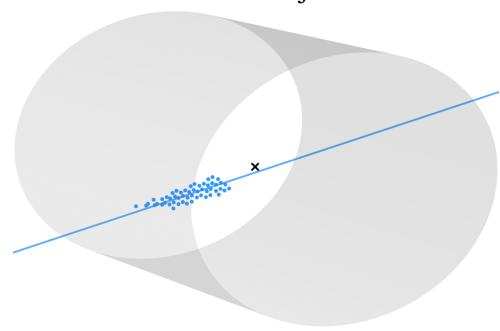
Para alguns eventos foi possível observar as trajetórias do feixe e partículas produto da reação saindo de um mesmo vértice (ver figura 6.1a). Neste caso, o vértice corresponde ao ponto médio da interseção dessas trajetórias. Porém, na maioria dos casos não foi possível observar a trajetória do feixe por causa do baixo ganho nos pixels centrais do detector. Por exemplo, a figura 6.1b mostra só dois produtos de reação saindo do mesmo vértice. No entanto, o vértice pode ser reconstruído pela extrapolação das retas ajustadas às trajetórias das partículas. Por fim, o caso mais complicado é quando apenas foi identificada uma trajetória (ver figura 6.1c). Para reconstruir o vértice com apenas uma trajetória, é necessário assumir uma direção para a partícula do feixe. Assim como foi mencionado no capítulo 5, se assume que o feixe tem direção $(0,0,1)$ e ponto de corte $(-3.4,-0.9,0)$ que foi calculado usando a posição média da projeção dos pontos de um conjunto de eventos no plano x - y . Portanto, o vértice corresponde ao ponto mais próximo da trajetória partícula com a trajetória assumida para o feixe.



(a) Exemplo de reação nuclear onde foram identificadas três trajetórias.



(b) Exemplo de reação nuclear onde foram identificadas duas trajetórias.



(c) Exemplo de reação nuclear onde foi identificada apenas uma trajetória.

Figura 6.1: Exemplos de reações nucleares com as trajetórias identificadas. O vértice de reação é indicado por um X, em preto. As cores servem apenas para a diferenciação.

Além da trajetória, a energia das partículas também é um parâmetro importante para a classificação e determinação da reação nuclear.

6.1.2 Reconstrução da energia das partículas

Numa primeira aproximação, a energia da partícula associada a uma determinada trajetória pode ser estimada a partir da equação 5.8, onde são somadas todas as cargas dos pontos da trajetória. Porém, a eficiência de detecção depende da energia depositada pela partícula no gás. Da mesma forma, alguns pontos das trajetórias das partículas podem estar ausentes devido ao processamento eletrônico ou mesmo na reconstrução das nuvens de pontos. Outra forma de obtermos (determinarmos) a energia da partícula é a partir da relação do comprimento da trajetória e a energia perdida pela partícula dentro do gás (poder de freamento ou stopping power). A perda de energia da partícula no gás depende principalmente da carga da partícula incidente e das propriedades do gás, como a pressão e temperatura. Essa energia perdida no gás foi calculada usando o programa LISE++

[75] e os respectivos comprimentos das trajetórias associadas a partículas como ^{16}O , alfas e prótons num gás de ^4He à uma pressão de 350 Torr. Partículas diferentes perdem energias de forma diferente e, portanto, têm alcances diferentes dentro do gás em função de sua energia. A figura 6.2 mostra o alcance em função da energia do ^{17}F , ^{16}O e próton, onde é possível ver a diferença de alcance das partículas dentro gás, especialmente entre as partículas pesadas (^{17}F , ^{16}O) e o próton.

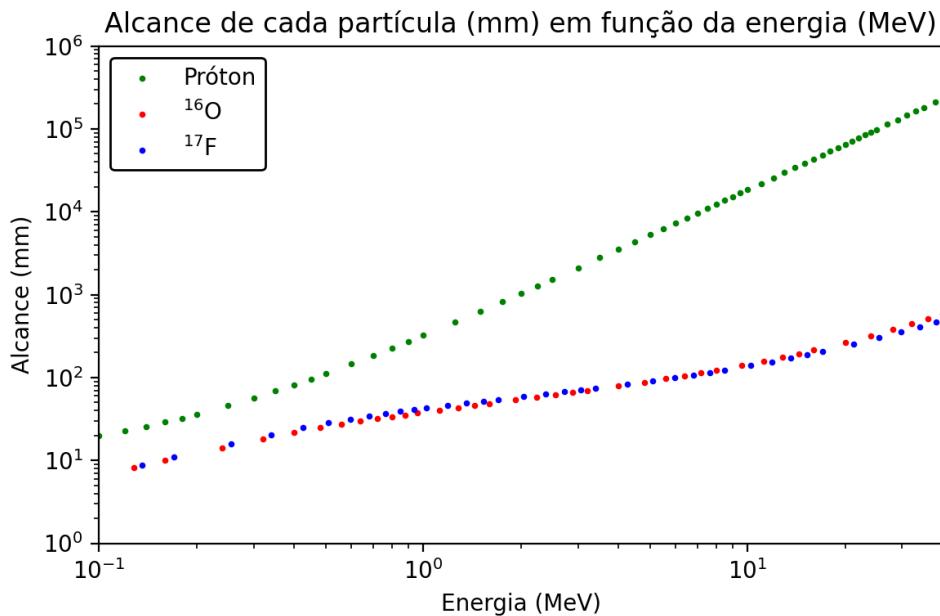


Figura 6.2: Alcance (mm) em função da energia (MeV) para o ^{17}F , ^{16}O e próton. A grande diferença está no próton que tem um alcance muito maior que os outros núcleos.

Por exemplo, para determinar a energia cinética do ^{17}F antes da reação nuclear, pode se usar a distância entre a janela do detector e o vértice da reação. Para isso, o espaço no eixo z foi discretizado, a fim de calcular a energia na posição da reação. O feixe, ao entrar no gás, possui energia inicial $E_0 = 34.76 \text{ MeV}$, e perde energia ao atravessar o gás. A energia de reação E_r é dada por

$$E_r = E_0 - E_{loss}, \quad (6.1)$$

onde E_{loss} é a energia do ^{17}F depositada no gás que é calculada a partir da distância L até o vértice da reação. O comprimento L é transformado em energia cinética usando uma tabela de *stopping power*. Similarmente, a energia de outras partículas (e.g. alfa), que param completamente no gás, é calculado com o alcance da partícula (relativo ao vértice) e a sua respetiva tabela de *stopping power*.

6.1.3 Determinação dos ângulos de reação

Como mencionamos anteriormente, uma reação nuclear pode ser investigada a partir de sua distribuição angular. Portanto, um dos parâmetros importantes para a determinação da distribuição angular de uma determinada reação é exatamente o ângulo em que as partículas são emitidas.

Para determinar os ângulos azimutal e axial (em coordenadas esféricas), foram feitos os ajustes tridimensionais dos aglomerados identificados (usando estimadores robustos ou algoritmos de clustering, como discutido no capítulo 5).

O ângulo de reação, para cada trajetória identificada, depende da determinação do versor de cada uma das trajetórias em relação ao versor do feixe. A partir das propriedades geométricas das trajetórias (versores das retas), são calculados os ângulos azimutal e axial com relação à direção em que o feixe incide na câmara (veja que a direção do feixe não necessariamente coincide com o eixo simétrico do TPC).

Para cada trajetória (reta) i , o ângulo azimutal ϕ_i é dado por

$$\phi_i = \text{atan2} \left(\frac{y_i}{x_i} \right), \quad (6.2)$$

onde atan2 é o arco-tangente calculado entre $-\pi$ e π , y_i e x_i são componentes x e y do versor da reta i . Para o ângulo polar θ , ele é calculado a partir do produto interno entre o versor da reta e o versor do feixe, ou seja:

$$\theta = \arccos \left(\frac{\vec{V}_i \cdot \vec{V}_f}{|\vec{V}_i| |\vec{V}_f|} \right), \quad (6.3)$$

onde \vec{V}_i é o versor da reta i e \vec{V}_f é o versor do feixe.

6.2 Identificação do canal de reação de breakup

A grande vantagem do alvo ativo é a possibilidade de detectarmos partículas provenientes das mais diversas reações nuclear. Por outro lado, isso dificulta consideravelmente a análise pois precisamos identificar e classificar essas reações. Por exemplo, ao graficarmos o comprimento do alcance das partículas em função do ângulo, temos a presença de todas as partículas. Isso fica evidente ao olhar para a figura 6.3, que mostra um histograma bidimensional do comprimento da trajetória da partícula em função do ângulo de espalhamento (polar) no referencial de laboratório. Eventos de espalhamento em ângulos menores que 5 graus e reações na janela de entrada foram previamente removidos neste

histograma.

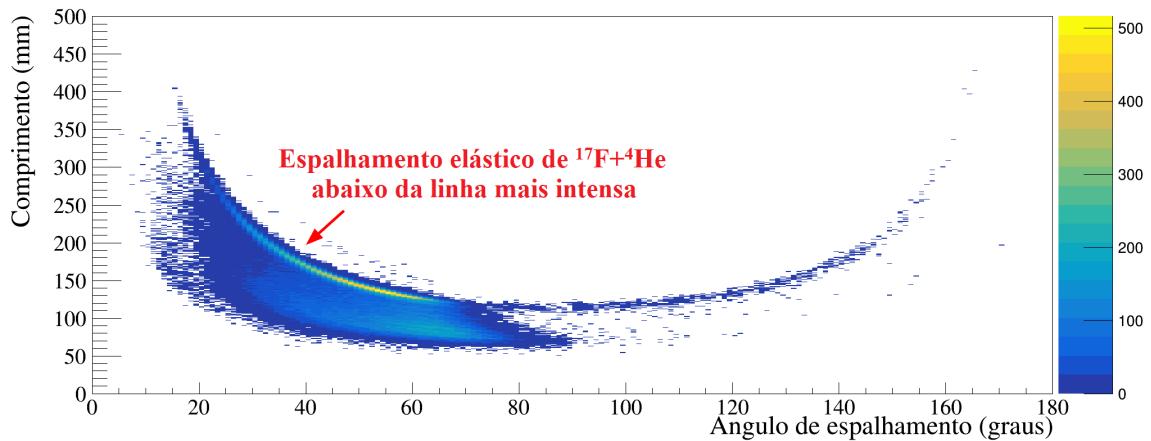


Figura 6.3: Histograma de comprimento da trajetória no eixo y e ângulo de espalhamento no eixo x .

Pela figura 6.3, é possível perceber dois aspectos importantes sobre os dados. O primeiro é que o canal de reação correspondente ao espalhamento elástico e inelástico, é predominante no espectro. Devido ao fato do experimento ter sido realizado em cinemática inversa, o ângulo de espalhamento da partícula recuo depende do valor Q da reação. Por exemplo, partículas alfa espalhadas elasticamente são detectadas em ângulos dianteiros menores que 90 graus. Os ângulos maiores que 90 graus neste caso correspondem a reações com $Q > 0$, por exemplo a reação (α, p) . O segundo aspecto é a ausência de eventos abaixo de 50 mm, como é possível ver na figura 6.3. Essa ausência ocorre devido ao baixo ganho da região central do detector micromegas que limita a detecção de tracks menores do que 60 mm. Da mesma forma, a ausência de eventos na parte superior do histograma corresponde a limitação geométrica do TPC que possui um raio de 150 mm e comprimento de 500 mm.

O foco principal desse trabalho é identificar e selecionar os eventos correspondentes ao canal de breakup $\alpha(^{17}\text{F}, {}^{16}\text{O} + p)$. Esse canal pode ser identificado a partir dos eventos com emissão de prótons. A figura 6.4 mostra um exemplo de breakup de ^{17}F em ${}^4\text{He}$ observado no experimento.

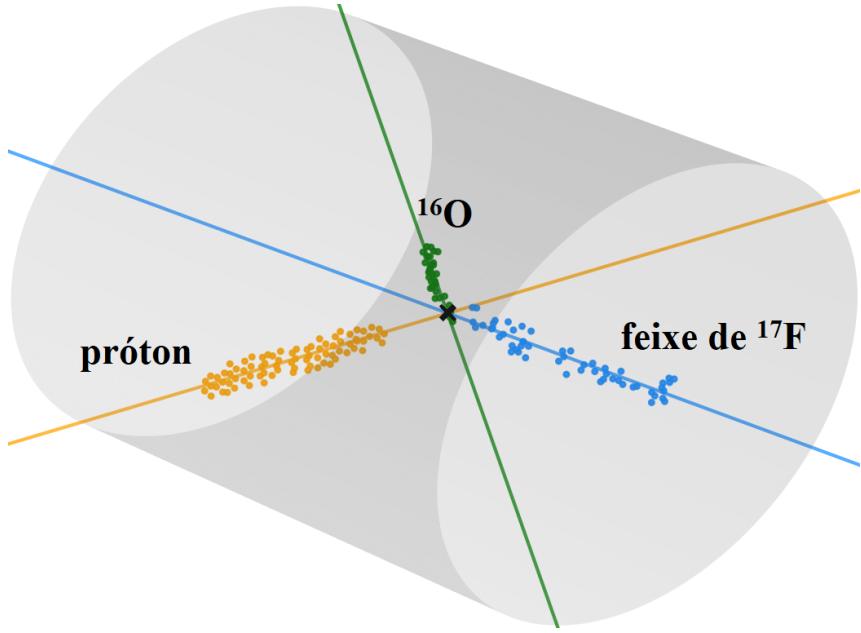


Figura 6.4: Evento onde há a trajetória originada por um próton (trajetória em amarelo), onde percebe-se que seu comprimento vai até o limite geométrico da câmara (cilindro cinza transparente). O feixe está em azul e o ^{16}O originou a trajetória em verde. É notável neste exemplo que a trajetória do próton é bem maior que o fragmento ^{16}O .

A seleção de eventos é feita aplicando vários gates (condições lógicas) para identificar os eventos com prótons. Isto pode ser feito num histograma da carga associada a uma trajetória em função de seu comprimento. A relação entre essas duas variáveis se dá por meio da expressão para o stopping power, onde o termo dominante da equação de Bethe Block [76] em baixas energias pode ser aproximado a

$$\frac{dE}{dx} \propto -\frac{4\pi Z^2 e^4 N_e}{m_e v^2}, \quad (6.4)$$

onde E é a energia da partícula com carga Ze (e é a carga do elétron) e velocidade v ao atravessar um gás com N_e elétrons por unidade de volume. Sabendo que $2E = Mv^2$, onde M é a massa da partícula, segue que

$$\int_0^E |2E'dE'| \propto \int_0^L \frac{4\pi Z^2 e^4 N_e M}{m_e} dx, \quad (6.5)$$

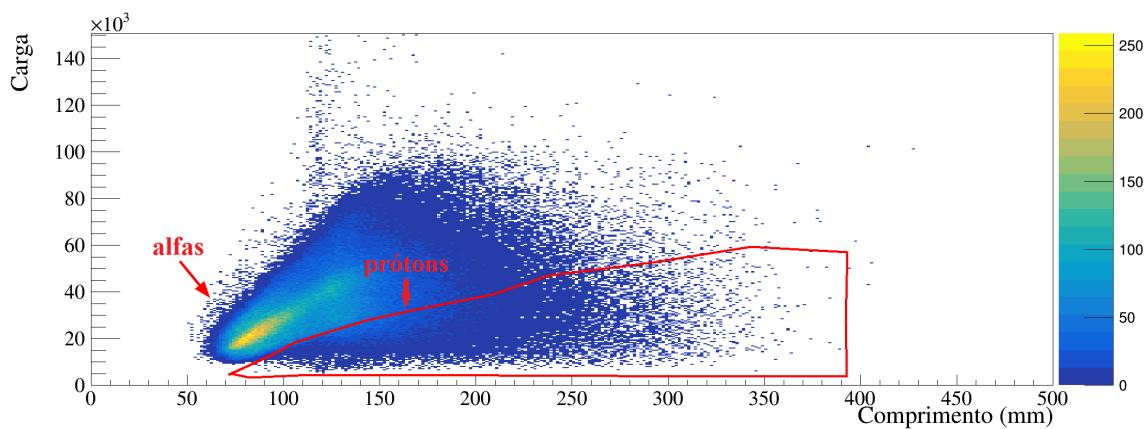
cuja integral resulta em

$$E^2 \propto (LM) \frac{4\pi Z^2 e^4 N_e}{m_e}, \quad (6.6)$$

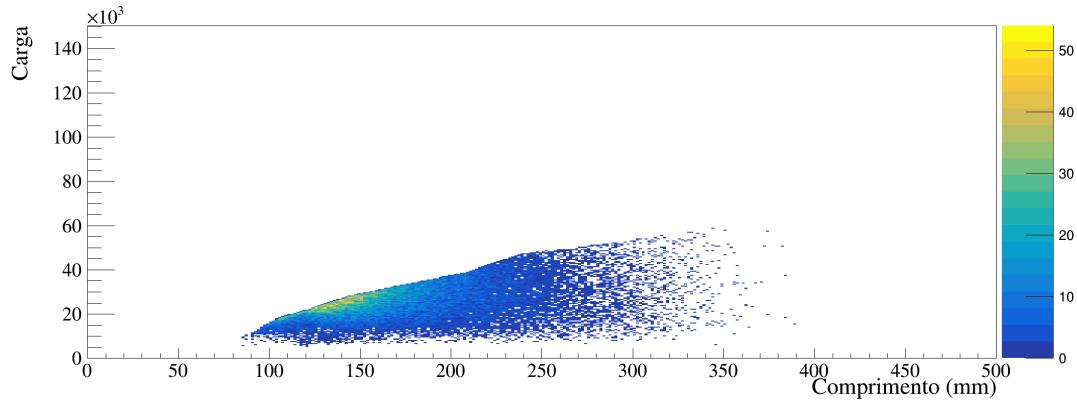
onde chegamos em

$$E \propto \sqrt{(LM)} \sqrt{\frac{4\pi Z^2 e^4 N_e}{m_e}}. \quad (6.7)$$

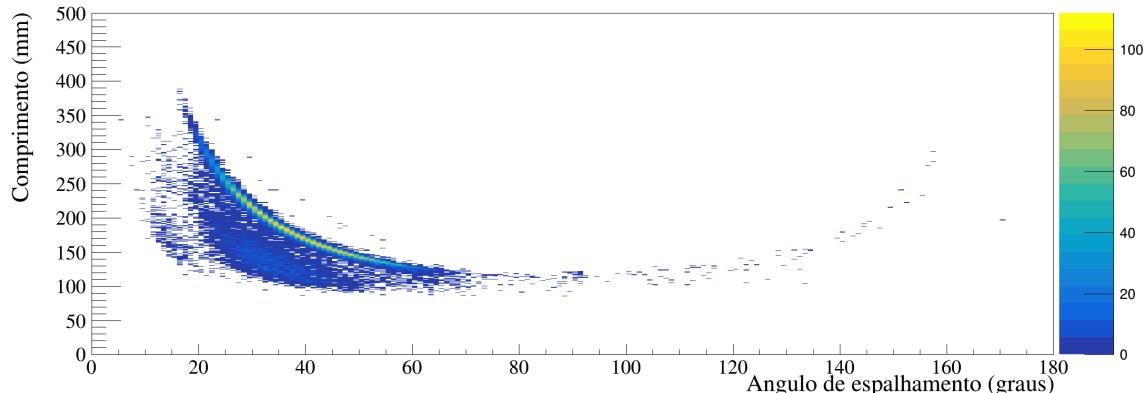
Portanto, existe uma relação do quadrado da energia com o comprimento da trajetória. O termo M mostra que quanto maior a massa M (e também o Z) da partícula, maior é a carga depositada. Dessa forma, os prótons podem ser identificados num histograma de carga (proporcional à energia depositada) em função do comprimento da trajetória como a região inferior à faixa das alfas, assim como mostrado na figura 6.5a. O resultado da aplicação do gate (polígono em vermelho na figura 6.5a que representa a região onde há as trajetórias dos prótons) para o espectro de comprimento em função do ângulo de espalhamento pode ser visto na figura 6.5c. Esse espectro deve ser comparado com o da figura 6.3 (sem aplicação do gate).



(a) Histograma bidimensional da carga em função do comprimento (mm). A seta vermelha à esquerda indica a região mais intensa, onde estão os eventos de espalhamento elástico (onde estão as trajetórias das partículas alfa). A seta à direita indica a região onde há as trajetórias dos prótons, que estão dentro da região onde há o polígono vermelha, que indica o gate, onde pontos (trajetórias) fora da região foram descartados.



(b) Histograma bidimensional da carga em função do comprimento (mm) após a aplicação do gate.



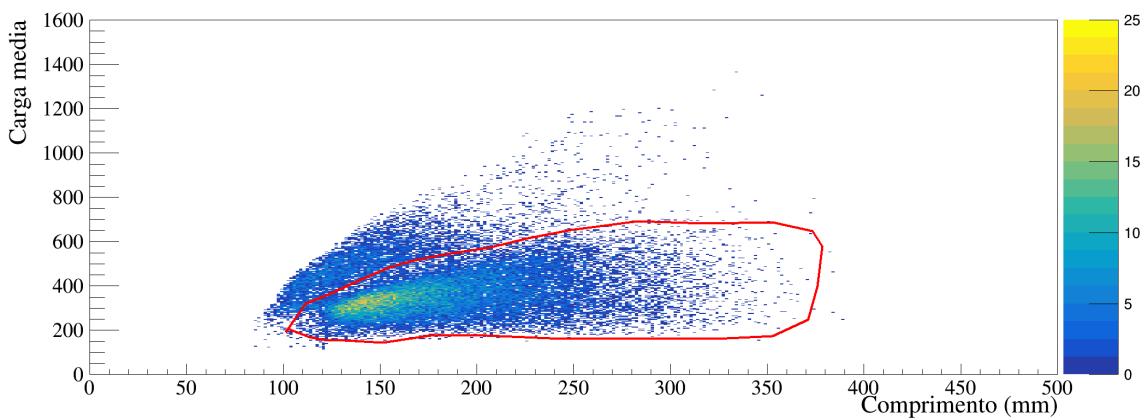
(c) Histograma bidimensional do comprimento (mm) em função do ângulo de espalhamento (graus) no referencial do laboratório após o gate.

Figura 6.5: Processo de aplicação do primeiro gate para identificar as trajetórias de reação de breakup. O resultado é o espectro da figura 6.5c.

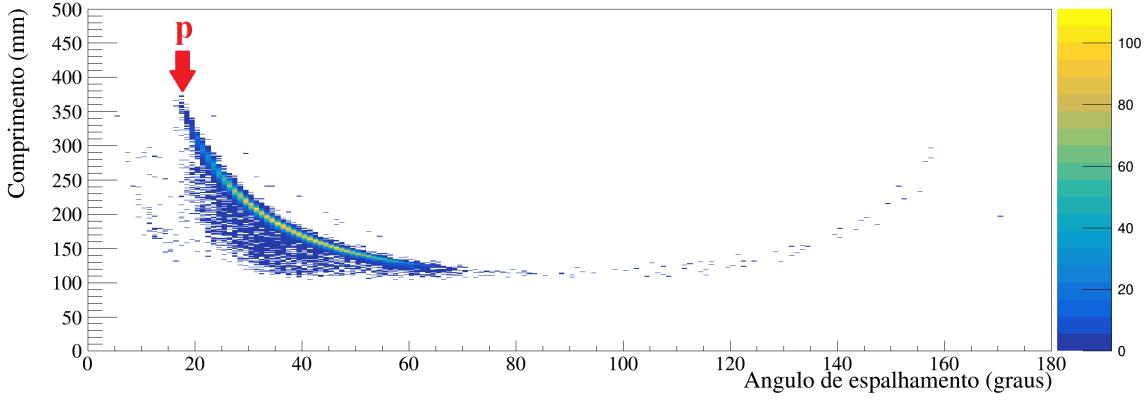
Como pode-se perceber, o gate removeu quase todos os eventos de espalhamento

elástico e inelástico de alfas. Somente sobraram duas faixas que correspondem às partículas que não param completamente no gás e partículas espalhadas em ângulos maiores que 20 graus que param no gás. A primeira faixa, mais intensa, corresponderia principalmente aos prótons e alfas, pois com apenas 1 MeV de energia cinética atravessam o gás (ver figura 6.2). A segunda faixa (no caso um aglomerado) seriam partículas alfas que pararam perto do limite geométrico do alvo ativo.

O gate anterior precisa ser refinado pois ainda há a contribuição de partículas alfa no espectro. Devido ao fato das partículas alfa depositarem maior energia num comprimento menor do que os prótons, a carga média depositada será maior para as partículas alfa. A carga média é definida como a carga total depositada pela partícula dividida pelo número total de pontos na trajetória. O espectro da figura 6.6a mostra a carga média em função do comprimento da trajetória usando o gate da figura 6.5. Pode-se perceber que tem dois aglomerados, um na parte inferior com maior densidade (prótons) e outro na parte superior, logo acima da linha vermelha na imagem, onde há partículas alfas, cujas trajetórias possuem uma carga média maior que a dos prótons. Ao excluir trajetórias fora da região delimitada pela linha vermelha na figura 6.6a, é possível identificar os prótons no espectro de comprimento em função do ângulo de espalhamento, mostrado na figura 6.6b.



(a) Histograma bidimensional da carga média em função do comprimento (mm). O polígono vermelha indica o segundo gate, onde eventos com trajetórias fora da região foram descartados.



(b) Histograma bidimensional do comprimento (mm) em função do ângulo de espalhamento (graus) após a aplicação do gate mostrado na figura 6.6a. Com esse gate, é possível identificar as trajetórias do próton, indicadas na figura.

Figura 6.6: Processo de aplicação do segundo gate para identificar os prótons. O resultado é um espectro da figura 6.6b.

Os eventos de breakup podem agora ser identificados a partir desses prótons. Para tanto, podemos simplesmente aplicar a esses prótons a condição de que devam corresponder a duas trajetórias coincidentes (excluindo o feixe) com o mesmo vértice. Essa condição elimina outros canais de reação, como o canal de fusão. Aplicadas essas condições, tem-se o espectro do comprimento em função do ângulo de espalhamento para eventos com duas trajetórias coincidentes, mostrado na figura 6.7.

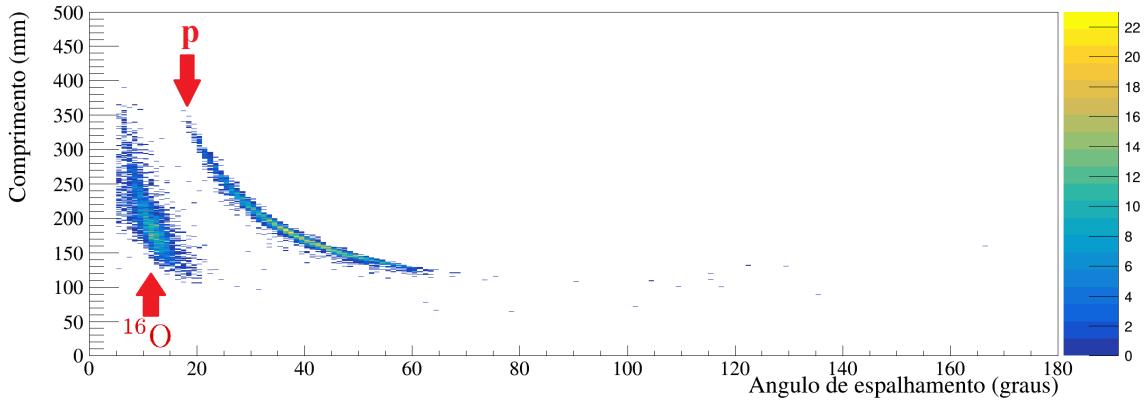


Figura 6.7: Histograma bidimensional do comprimento (mm) em função do ângulo de espalhamento (graus) para eventos com coincidência. Nele é possível identificar os prótons (trajetórias que não param no gás) e o ^{16}O , que possui trajetórias com ângulos mais direcioneiros e comprimentos menores.

A partir da identificação do canal de breakup nos eventos, foi possível construir as distribuições angulares da reação de breakup do ^{17}F em ^4He .

6.3 Construção das distribuições angulares

Identificado o canal de breakup, foi possível construir as distribuições angulares para essa reação. Devemos ainda ressaltar que os eventos para os quais foram identificadas uma única partícula, no caso o próton, são chamadas de breakup inclusivo. Quando há coincidência, ou seja, a medida coincidente do próton e do ^{16}O , é chamada de breakup exclusivo.

Com o alvo ativo é possível medir reações em diversos ângulos e energias. Portanto, para obtermos as distribuições angulares é preciso primeiro discretizar (colocar em bins) os ângulos de espalhamento θ_p (ângulos de espalhamento dos prótons no referencial do laboratório). Escolhemos um intervalo angular de 4 graus para esse *binning* para minimizar as flutuações estatísticas na distribuição angular. A incerteza sobre essa escolha é de aproximadamente 1°. A seção de choque diferencial $\frac{d\sigma}{d\Omega}(\theta_p)$ é dada por [19]

$$\frac{d\sigma}{d\Omega}(\theta_p) = \frac{Y}{N_b N_t \Delta\Omega(\theta_p)} J_{\text{lab}} \times 10^{27} \text{ mb/sr.} \quad (6.8)$$

Os termos dessa expressão são:

- Y

É o número de contagens dos eventos de interesse. Esse valor é obtido a partir de

contagens de eventos em que foi identificada coincidencia, para o breakup exclusivo, ou foi identificado apenas o próton, para o breakup inclusivo.

- N_t

É o número total de partículas no alvo gasoso por unidade de área (perpendicular ao feixe). Esse número é obtido a partir da equação dos gases ideais [77], onde chega-se em:

$$N_t = \frac{PdN_A}{RT} \text{ part./cm}^2, \quad (6.9)$$

onde P é a pressão do alvo gasoso, d é o comprimento da câmara, N_A é o número de Avogrado, R a constante universal dos gases ideais e T a temperatura. Para $P = 350$ Torr e $T = 300$ K, $N_t = 5.63 \times 10^{20}$ part./cm².

- N_b

É o número de partículas do projétil que estão incidindo no alvo gasoso, determinado a partir de medidas de espalhamento elástico feitas com o próprio alvo gasoso ⁴He. O espalhamento elástico obtido experimentalmente (calculado pela equação 6.8) é comparado ao valor teórico para assim obter o número médio de partículas que estão incidindo no alvo:

$$N_b = \left(\frac{Y}{N_t \Delta\Omega} \right)_{\text{alvo}} \left(\frac{d\sigma}{d\Omega} \right)^{-1}_{\text{Ruth}}. \quad (6.10)$$

- $\Delta\Omega(\theta_p)$

Ângulo sólido obtido através do ângulo θ_p , dado por:

$$\Delta\Omega(\theta_p) = 2\pi \int_{\theta_p}^{\theta_p + \theta_{bin}} \sin(\theta) d\theta. \quad (6.11)$$

- J_{lab} Termo que é o Jacobiano para a transformação de coordenadas. Como estamos usando o ângulo θ_p no referencial do laboratório, então $J_{\text{lab}} = 1$.

A incerteza $\partial \left(\frac{d\sigma}{d\Omega} \right)$ da seção choque é dada por

$$\partial \left(\frac{d\sigma}{d\Omega} \right) = \left(\frac{d\sigma}{d\Omega} \right) \left[\left(\frac{\partial Y}{Y} \right)^2 + \left(\frac{\partial N_b}{N_b} \right)^2 + \left(\frac{\partial N_t}{N_t} \right)^2 + \left(\frac{\partial (\Delta\Omega)}{(\Delta\Omega)} \right)^2 \right]^{1/2}, \quad (6.12)$$

onde ∂x é a incerteza da respectiva variável x .

A normalização deve ser obtida a partir da análise do canal elástico do experimento. O valor da normalização foi determinado como sendo 0.243 (56), onde a luminosidade integrada L é

$$L = 2.43 \times 10^{26} \text{ cm}^{-2}. \quad (6.13)$$

Uma das grandes vantagens da utilização do alvo ativo em medidas de reações nucleares é a possibilidade de obtermos uma variação da energia do feixe conforme este vai perdendo energia ao penetrar no volume ativo do alvo gasoso.

Para construir a distribuição de energia consideramos a seção de choque para faixas discretas de energia. O tamanho do bin da energia foi escolhido de forma a evitar flutuações estatísticas, devido a poucas contagens em certas energias. Como a energia do feixe depende da posição em z do vértice de reação, ela foi discretizada a cada 10 mm, o que resulta em bins de aproximadamente 180 keV (no referencial do centro de massa) de largura. A partir dessas bins, foi calculada a energia do feixe. A figura 6.8 mostra o espectro da seção de choque em função da energia no referencial do centro de massa. A seguir as distribuições angulares para a reação de breakup (seção de choque em função do ângulo de espalhamento do próton) foram determinadas para várias energias do feixe, conforme mostradas nas figuras 6.9, 6.10 e 6.11.

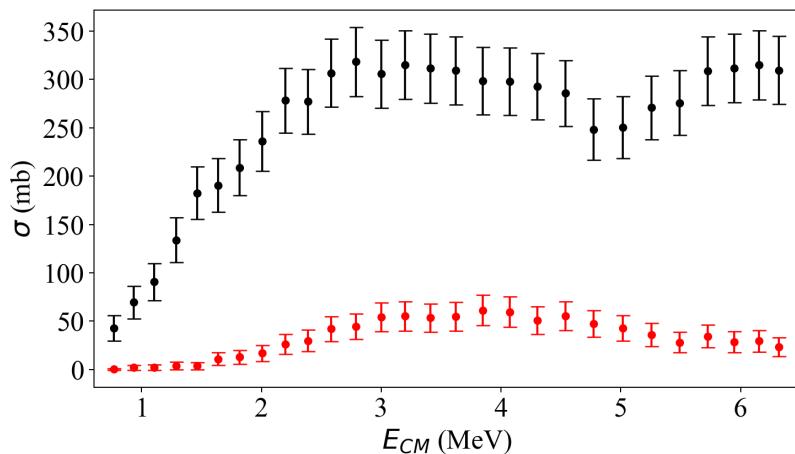


Figura 6.8: Gráfico que mostra a seção de choque σ em função da energia no referencial do centro de massa. Os pontos de cor preta correspondem à seção de choque do breakup inclusivo e os pontos de cor vermelha correspondem à seção de choque do breakup exclusivo.

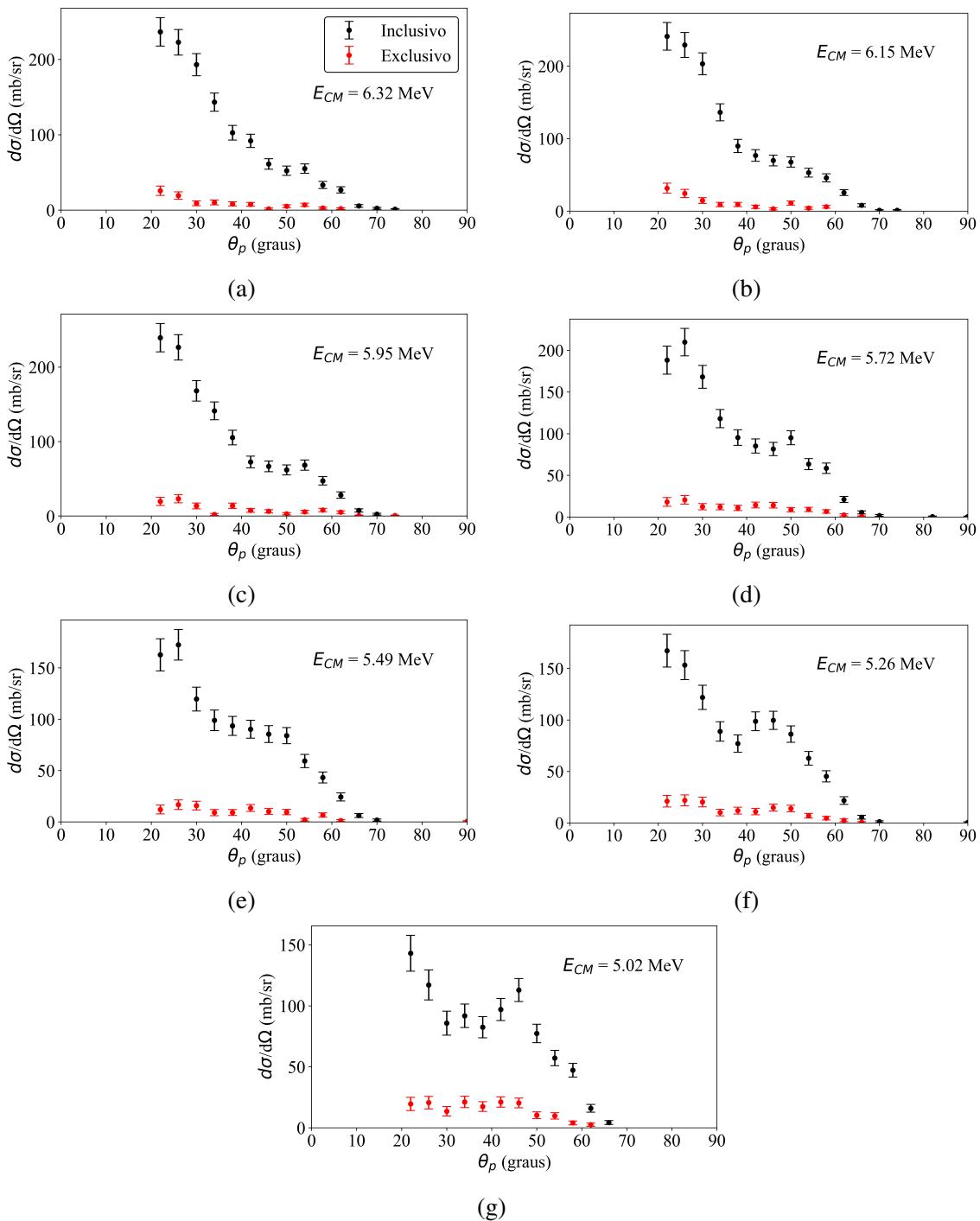


Figura 6.9: Distribuições angulares de breakup do ^{17}F em ^4He em função do ângulo de espalhamento do próton θ_p no referencial do laboratório para várias energias do feixe. As energias estão acima da barreira de Coulomb da reação (4.2 MeV).

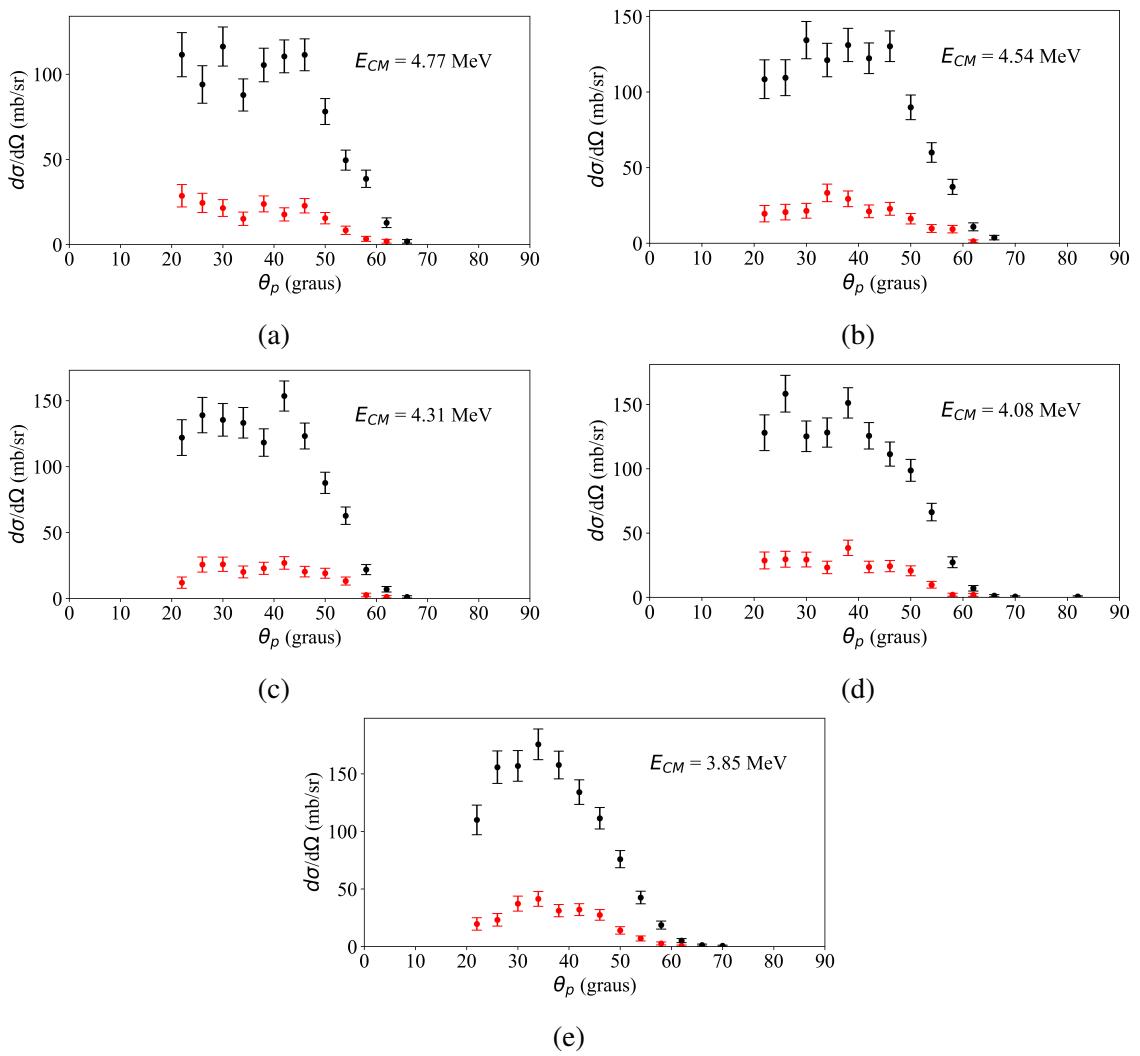
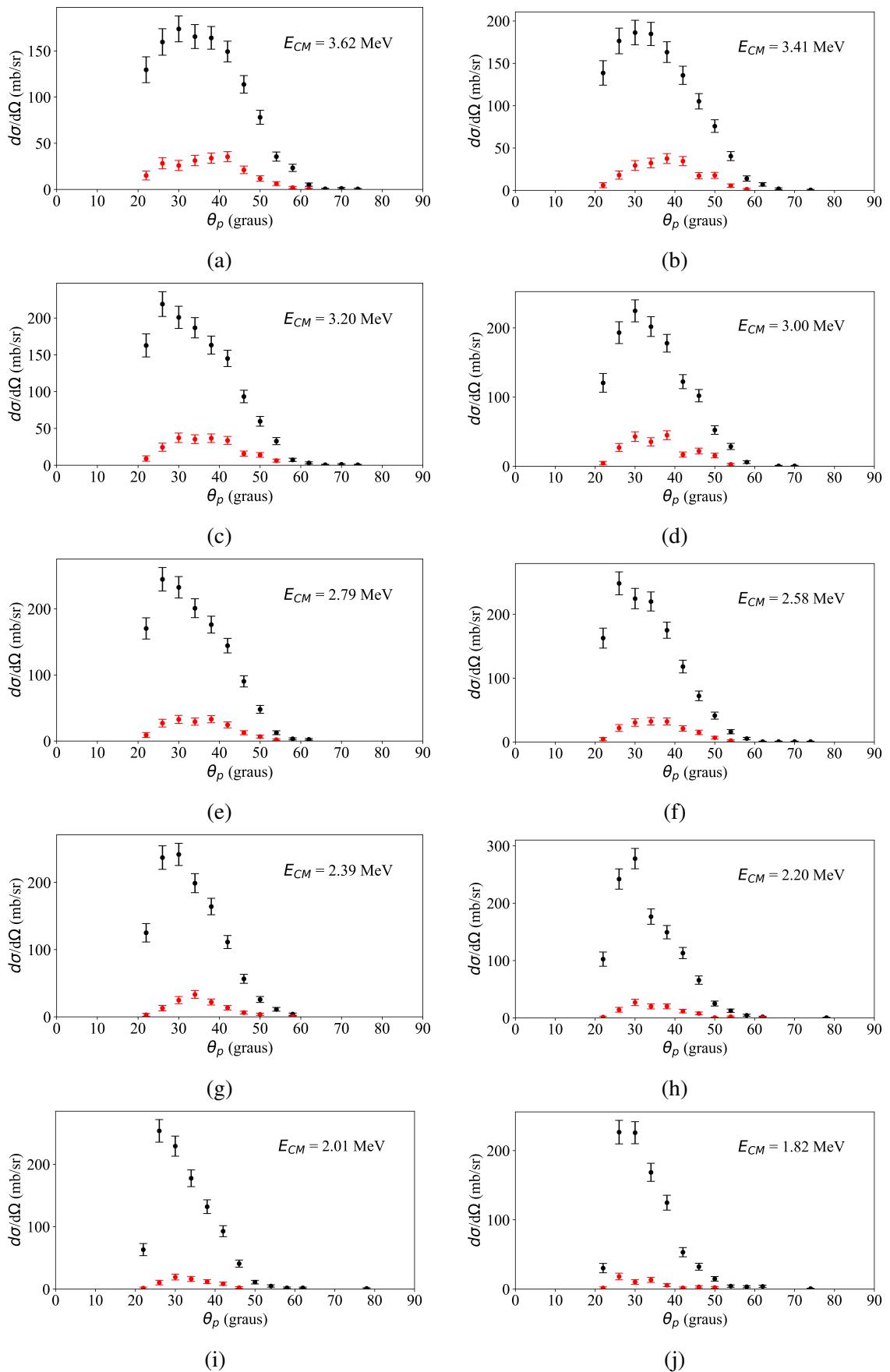


Figura 6.10: Distribuições angulares de breakup do ^{17}F em ^4He em função do ângulo de espalhamento do próton θ_p no referencial do laboratório para várias energias do feixe. As energias estão próximas da barreira de Coulomb da reação (4.2 MeV).



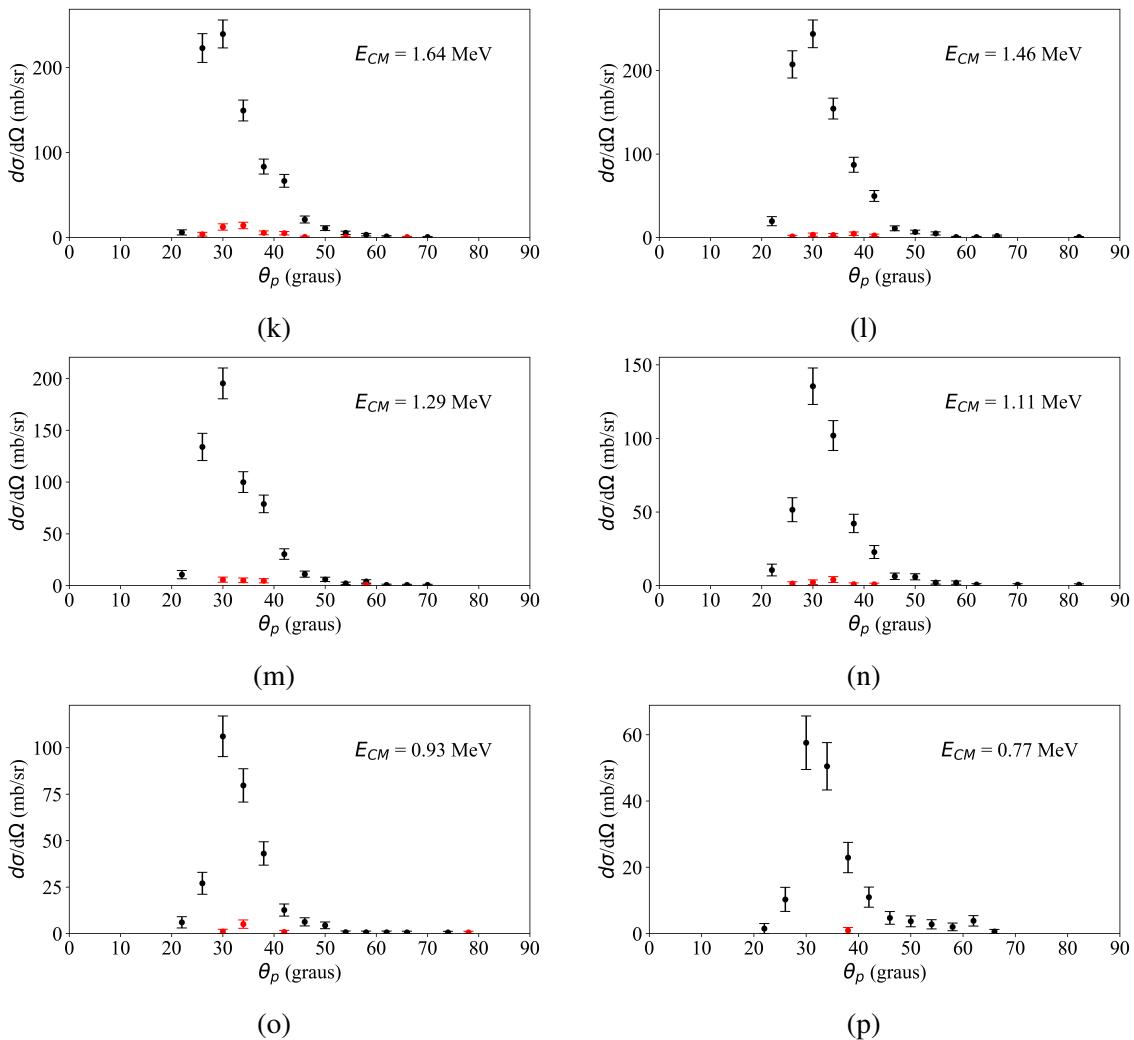


Figura 6.11: Distribuições angulares de breakup do ^{17}F em ^4He em função do ângulo de espalhamento do próton θ_p no referencial do laboratório para várias energias do feixe. As energias estão abaixo da barreira de Coulomb da reação (4.2 MeV).

6.4 Análise e comparação dos resultados

Os resultados das figuras 6.9, 6.10 e 6.11 mostram distribuições com energias acima (figura 6.9), próximas (figura 6.10) e abaixo (figura 6.11) da barreira de Coulomb da reação ($V_B = 4.2$ MeV). As distribuições angulares para o processo de breakup foram determinadas para energias acima (figura 6.9) próximas (figura 6.10) e abaixo (6.11) abaixo da barreira coulombiana para o sistema $^{17}\text{F} + ^4\text{He}$. ($V_B = 4.2$ MeV) As distribuições entre 3.63 e 4.77 MeV se distribuem de modo horizontal em ângulos entre 20° e 50° . Distribuições acima da barreira possuem valores maiores em ângulos mais dianteiros (de 20° à 30°). Embaixo da barreira as distribuições são mais estreitas e parecem ter um valor máximo em 30° .

Uma análise teórica dessas distribuições angulares para reações de breakup (dissociação coulombiana) envolvendo projéteis de núcleos ricos em nêutrons (fracamente ligados) usualmente se baseia em teorias de perturbação em primeira ordem. Para o breakup de núcleos ricos em prótons fracamente ligados, como o caso do ^{17}F , são sugeridos outros métodos para o cálculo da seção de choque [78, 79].

Investigação do breakup do projétil de ^{17}F incidindo em alvos mais pesados (^{58}Ni e ^{208}Pb) já foi realizada por outros autores recentemente [78]. As medidas foram feitas acelerando um feixe de ^{17}F a uma energia de 170 MeV, no referencial do laboratório, incidindo em um alvo de ^{58}Ni ($E_{\text{c.m.}} = 131.47$ MeV) e também em um alvo de ^{208}Pb ($E_{\text{c.m.}} = 157.16$ MeV). As distribuições angulares para cada reação estão na figura 6.12. Era esperado desse experimento que a seção de choque de breakup do ^{17}F fosse maior para o alvo de ^{208}Pb ($Z=82$) do que para o ^{58}Ni ($Z=28$), devido a uma maior contribuição da interação Coulombiana do ^{208}Pb . No entanto, foi observado o oposto. A seção de choque de choque do ^{17}F em ^{208}Pb foi menor que em ^{58}Ni , resultado que indicou uma possível interação entre as contribuições de Coulomb e nuclear para o breakup [78, 5]. Assim, investigar o breakup do ^{17}F com um alvo com Z ainda menor, como o caso do presente trabalho, pode ajudar a compreender melhor o mecanismo de breakup, melhorando o entendimento das contribuições, pois a interação de Coulomb é ordens de grandeza menor que os casos anteriores, favorecendo o estudo da contribuição da parte nuclear e também possíveis efeitos de interferência.

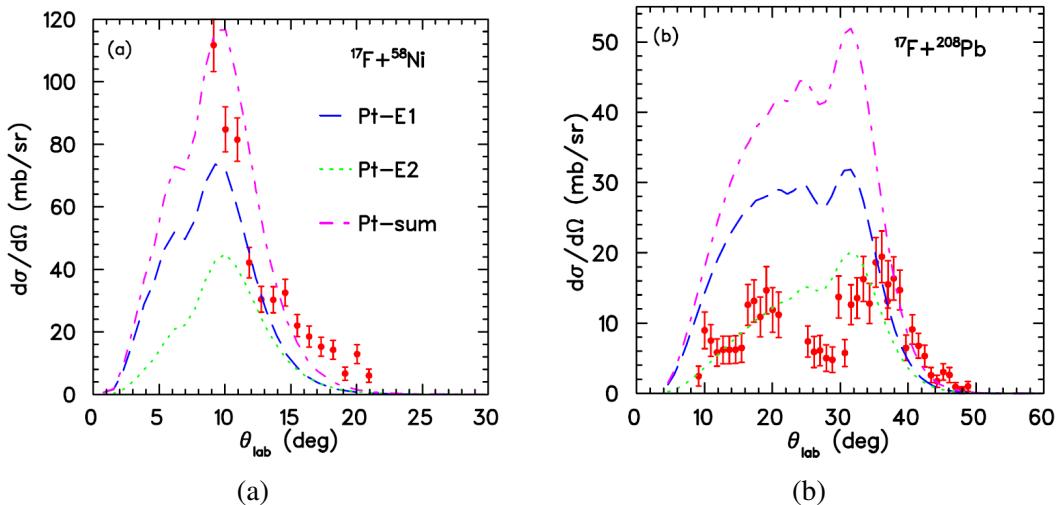


Figura 6.12: Distribuições angulares do breakup exclusivo para (a) $^{17}\text{F} + ^{58}\text{Ni}$ e (b) $^{17}\text{F} + ^{208}\text{Pb}$. As linhas tracejadas representam previsões teóricas para a dissociação de Coulomb a partir da teoria de perturbação em primeira ordem, onde Pt-E1 corresponde à transição E1, Pt-E2 à transição E2 e Pt-Sum é a soma das transições E1 e E2. Mais detalhes podem ser encontrados na Ref. [78], de onde foram extraídas as figuras.

O foco do presente trabalho foi obter experimentalmente as distribuições angulares para o breakup do $^{17}\text{F} + ^4\text{He}$. No entanto, podemos comparar as formas das distribuições angulares obtidas para energias acima da barreira (figura 6.9) com a forma das distribuições angulares para os sistemas $^{17}\text{F} + ^{58}\text{Ni}$. Comparando o comportamento em ângulos dianteiros, onde há a predominância de pontos, as respectivas seções de choque são maiores em ^4He que em ^{58}Ni . Para energias próximas da barreira de Coulomb (entre 3.63 MeV e 4.77 MeV, figura 6.10), a distribuição apresentada na figura 6.12b é mais semelhante, onde não há um platô para ângulos entre 5° e 10° . O resultado importante é que as seções de choque em ^4He para energias acima da barreira são maiores que as seções de choque em ^{58}Ni e em ^{208}Pb . Cálculos de CDCC (Continuum-Discretized Coupled-Channels), onde são acrescentados estados do contínuo além dos estados ligados do ^{17}F , para os dados apresentados na figura 6.12, realizados na Ref. [5], mostram que há interações (interferências construtivas ou destrutivas) entre as diferentes contribuições para o breakup (contribuição de Coulomb e nuclear). A figura 6.13 mostra os cálculos das contribuições nuclear e de Coulomb para a seção de choque bem como o cálculo de CDCC com ambas as contribuições, onde é possível ver que o método descreve melhor os dados que o método anterior (teoria de perturbação em primeira ordem).

Outro experimento do breakup do ^{17}F em ^{58}Ni , realizado recentemente [80], porém com energias próximas da barreira de Coulomb, cujas distribuições angulares estão na figura 6.14, também indicaram que novos cálculos (diferentes do método perturbativo) são necessários para a análise das seções de choque [80, 4]. Cálculos de CDCC indicaram novamente que há efeitos de interferência entre a contribuição de Coulomb e a contribuição nuclear para o breakup. No caso de energias próximas da barreira, a seção de choque diminui em cerca de 20%, algo notado nas distribuições com energias próximas da barreira, mostradas na figura 6.10.

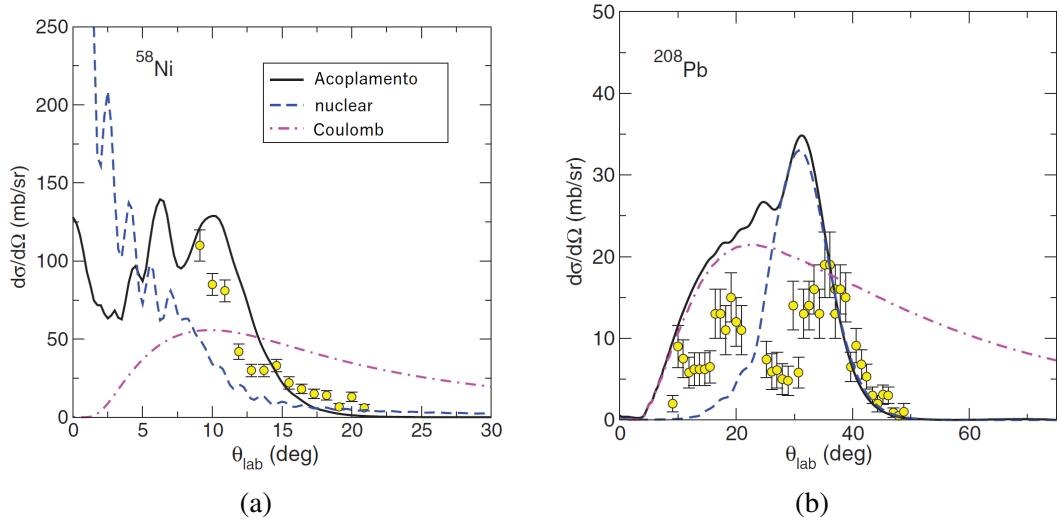


Figura 6.13: Cálculos de CDCC para as contribuições da seção de choque de breakup exclusiva. A linha tracejada é o cálculo da contribuição nuclear, a linha ponto-tracejada é o cálculo para a contribuição de Coulomb. A linha sólida preta é o cálculo de CDCC incluindo tanto a contribuição nuclear e de Coulomb. Mais detalhes podem ser encontrados em [5].

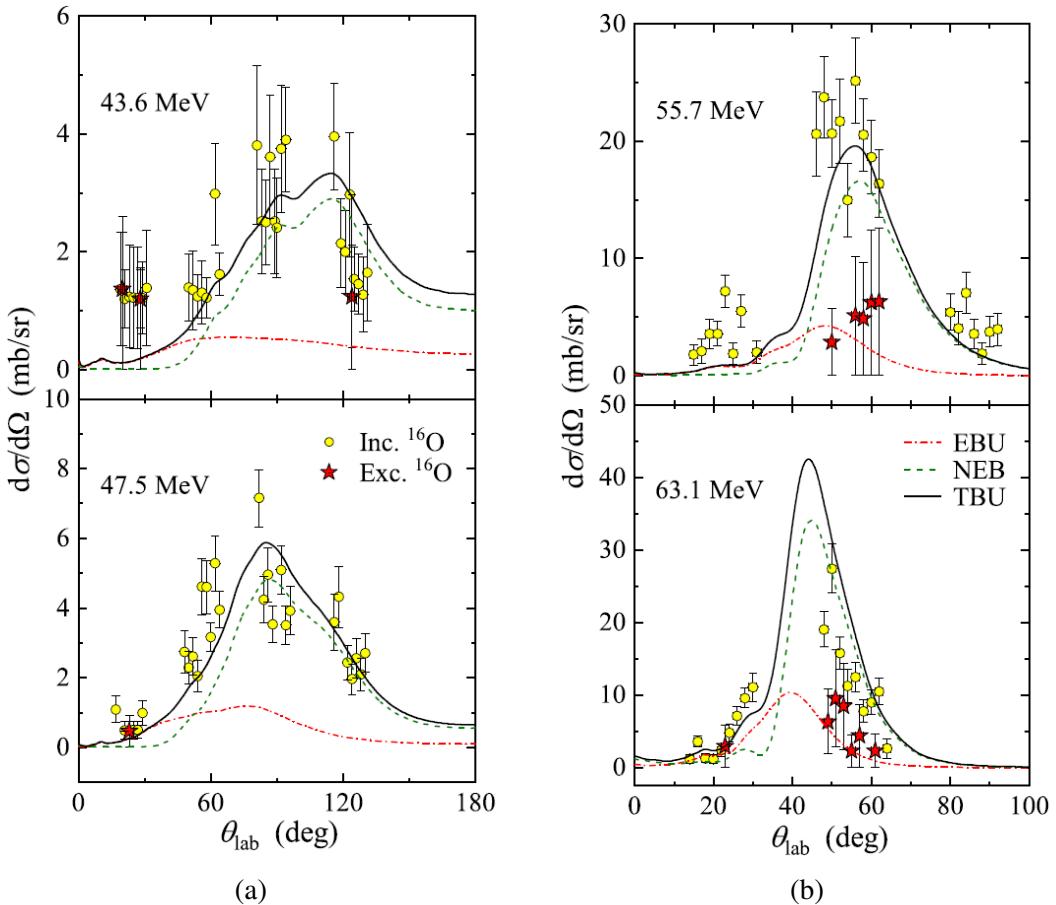


Figura 6.14: Distribuições angulares do breakup inclusivo (ponto amarelo) e exclusivo (estrela vermelha) para a reação de breakup $^{17}\text{F} + ^{58}\text{Ni}$ com energias próximas da barreira de Coulomb. A linha ponto-tracejada são cálculos de CDCC para o breakup elástico (EBU), a linha tracejada é o calculo de CDCC para o breakup não elástico (NEB) e a linha sólida é a soma de ambas (TBU). As energias em cada figura estão no referencial do laboratório. Mais detalhes podem ser encontrados na Ref. [80].

Os resultados dos cálculos de CDCC nos dados experimentais apresentados, mostram que existem interferências destrutivas e construtivas nas contribuições de Coulomb e nuclear do breakup. A contribuição parece ter uma relevância bem maior em núcleos com Z menor. Em núcleos com Z grande, como o caso do ^{208}Pb , os cálculos teóricos indicam que a interferência entre as interações nuclear e Coulombiana reduzem consideravelmente a seção de choque [4, 5]. Para o caso de energias próximas da barreira de Coulomb, a seção de choque é ainda menor, corroborando com os dados obtidos com o experimento desse trabalho. Com esses resultados, será possível entender melhor a natureza do breakup, qual relevância de cada contribuição, como os efeitos de interferência acontecem, em diversas regiões de energia (acima, próxima e abaixo da barreira de Coulomb).

Capítulo 7

Conclusões

Esse trabalho abordou um método experimental e análise de dados para a obtenção das distribuições angulares para a reação de breakup ${}^4\text{He}({}^{17}\text{F}, {}^{16}\text{O} + p){}^4\text{He}$. O experimento foi realizado com o alvo ativo pAT-TPC junto com o sistema TWINSOL (University of Notre Dame, Estados Unidos). A reação foi medida em cinemática inversa usando um feixe secundário de ${}^{17}\text{F}$ com energia de 34.46 MeV interagindo com um alvo gasoso de hélio puro.

Com esse sistema de detecção foi possível fazer a reconstrução tridimensional dos eventos (nuvens de pontos) a partir de milhões de sinais digitais geradas pelo detector. Neste trabalho, algoritmos baseados em Machine Learning foram desenvolvidos para a reconstrução e analise das nuvens de pontos. A reconstrução foi dividida em três etapas: remoção do fundo, deconvolução dos pulsos e identificação de picos. Para cada etapa foi construído um banco de dados para as redes convolucionais usando pulsos crus de entrada e a respetiva saída desejada para cada sinal. Isso permitiu que os algoritmos aprendessem de forma supervisionada os padrões dos dados experimentais. Os resultados das redes neurais tem a mesma qualidade do que algoritmos tradicionais usados para essas análises, mas o tempo de calculo é 90 vezes mais rápido com a técnica desenvolvida neste trabalho. Os algoritmos desenvolvidos neste trabalho correspondem à primeira aplicação prática de técnicas de machine learning numa analise de dados com alvo ativo.

Para a análise das nuvens de pontos foi necessário identificar as trajetórias tridimensionais das partículas dentro do gás. Isso foi feito através de algoritmos de *clustering* e estimadores robustos (pRANSAC) [14] que permitem identificar aglomerados mesmo na presença de ruído (*outliers*). Algumas alternativas para a analise de nuvens de pontos, como o *PointNet* [69], foram investigadas usando redes neurais. A rede neural PointNet pode ser usada tanto para a classificação global da nuvem de pontos, quanto para a segmentação semântica (classificação ponto a ponto). A rede foi testada para identi-

ficar quantas trajetórias existem nas nuvens de pontos e demonstrou bons resultados (boa acurácia). A segmentação semântica também foi utilizada para a identificação de outliers (globais e locais), mostrando resultados satisfatórios. Para uma abordagem mais direta ao problema, foi sugerido o uso da rede neural *ConstrastNet* [74] para estudar *clustering* usando métodos supervisionados como continuação deste trabalho. A rede será capaz de fazer o *clustering* nos dados, retornando as trajetórias de cada partícula num mesmo evento.

A partir da análise das nuvens de pontos foi possível reconstruir trajetórias das reações nucleares dentro do gás. Em particular, neste trabalho foi estudada a reação de breakup $^{17}\text{F}(^{4}\text{He}, ^{16}\text{O}+p)^{4}\text{He}$. O canal de breakup foi identificado a partir dos prótons produzidos pela interação do feixe ^{17}F em ^{4}He ($V_B = 4.2$ MeV). As distribuições angulares foram obtidas em energias de centro de massa entre 0.77 e 6.32 MeV. Os resultados foram comparados com experimentos realizados anteriormente usando alvos de ^{58}Ni e ^{208}Pb .

Os resultados deste trabalho mostram que a seção de choque em ^{4}He , próxima a barreira Coulombiana, é maior do que em ^{58}Ni e ^{208}Pb . Isto seria uma possível indicação de interferência entre as interações nuclear e Coulombiana que reduz o valor da seção de choque de breakup. Junto com colaboradores teóricos, serão realizados cálculos para entender os resultados experimentais apresentados neste trabalho.

Apêndice A

Aplicações de machine learning na física nuclear

Em física nuclear, o uso de técnicas de machine learning tem se mostrado cada vez mais importante. Na continuação são citados alguns exemplos.

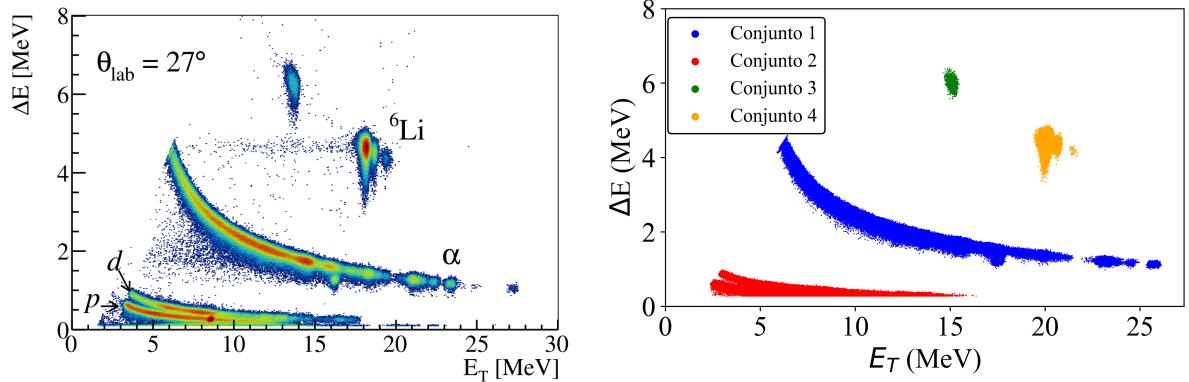
A.1 Análise de espectros para identificação de partículas (*particle identification, PID*)

O uso de técnicas de aprendizado não supervisionado pode ser usado para a identificar de partículas em espectros $\Delta E - E$. Neste tipo de espectros, as partículas são identificadas e separadas por regiões que facilmente podem ser identificadas visualmente [81]. No entanto, a identificação visual desses espectros com os sistemas de detecção com milhares de canais é inviável de forma manual. Assim, ferramentas de machine learning (ou inteligencia artificial) são de grande relevância para esse tipo de analise.

Os resultados apresentados nessa seção foram obtidos na etapa inicial desta dissertação. Para o presente estudo, o espectro biparamétrico ($\Delta E-E$) mostrado na Figura A.1a foi analisado. Os dados correspondem a reação de espalhamento e transferência no sistema ${}^6\text{Li} + {}^{12}\text{C}$.

O objetivo é identificar aglomerados (ou *clusters*) diferentes a partir do espectro. É claro para o olho humano que existem conjuntos diferentes, e eles podem ser identificados usando algoritmos de aprendizado não supervisionado, como por exemplo o *density-based spatial clustering of applications with noise* (DBSCAN) [64]. O DBSCAN consegue identificar clusters apenas com as informações características de densidade existente pelos conjuntos que existem nos dados. O resultado da aplicação do algoritmo está

na figura A.1b, onde é mostrado que o espectro agora está separado por diferentes conjuntos.



(a) Espectro biparamétrico $\Delta E \times E$ (energia no referencial do centro de massa).

(b) Espectro biparamétrico $\Delta E \times E$ (energia no referencial do centro de massa) com os conjuntos identificados com o uso do DBSCAN.

Figura A.1: Espectros biparamétricos $\Delta E \times E$, em A.1a o espectro cru com uma pré marcação das partículas, e em A.1b os conjuntos identificados usando machine learning.

Usando o conjunto 1, por exemplo, é possível criar o histograma de identificação de partículas dado pela figura A.2.

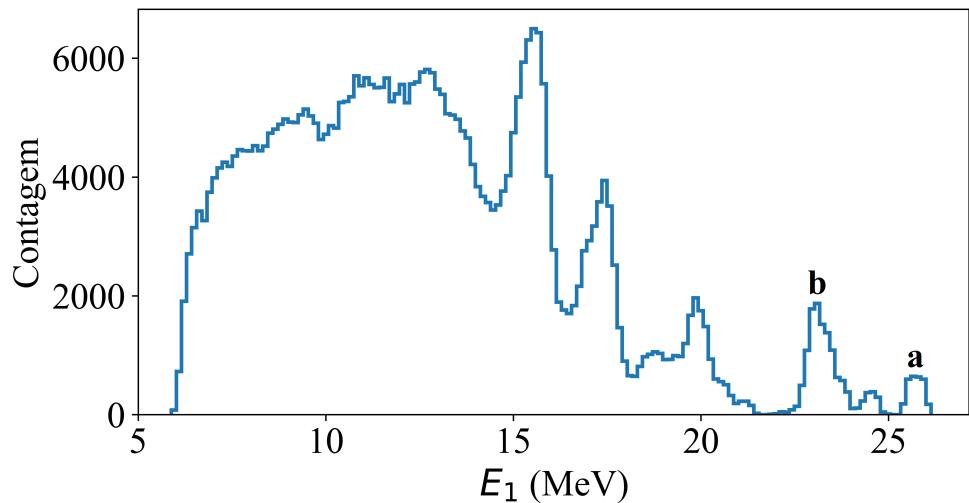


Figura A.2: Histograma de identificação de partículas, a partir do conjunto 1 identificado na figura A.1b. Em **a** temos o primeiro pico que corresponde ao estanho fundamental do ${}^{14}\text{N}$ e em **b** temos o primeiro estado excitado do ${}^{14}\text{N}$.

A.2 Estimativa de raios e massas nucleares

Frequentemente é necessário calcular com alta precisão observáveis que ainda não foram medidos, para contribuir com dados já existentes. É possível estimar propriedades de núcleos usando modelos e dados experimentais já existentes em conjunto com técnicas de machine learning.

É possível usar redes neurais que façam previsão de massa nucleares com informações de massa já disponíveis. Isso foi feito usando o algoritmo *Light Gradient Boosting Machine* (LightGBM) [82]. O algoritmo é uma rede neural cujo aprendizado é supervisionado e minimiza o erro entre a energia de ligação teórica e a energia de ligação experimental dos núcleos (essa diferença é chamada de resíduo), usando 10 quantidades físicas como dados de entrada (input) [83]. O desvio quadrático médio para a massa dos núcleos é de 0.234 ± 0.022 MeV, valor melhor que muitos modelos físicos de massa nuclear. A figura A.3 mostra os resíduos calculados entre a energia de ligação calculada pelo rede neural e a energia de ligação determinada experimentalmente.

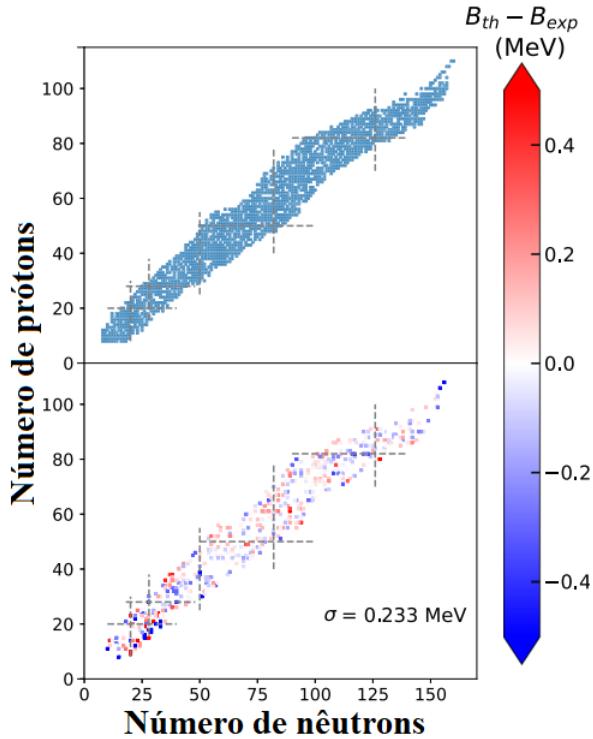


Figura A.3: Painel acima mostra a localização dos núcleos usados para o treino da rede neural. No painel de baixo temos o erro entre a previsão e o valor experimental da energia de ligação do núcleo para os núcleos usados como dados de validação. σ é o erro quadrático médio da rede neural.

Outro exemplo de uso é para a estimativa de raio de carga nuclear. Usando uma rede

neural (chamada de *Bayesian neural network extended liquid drop* - BNN-ELD) FC com duas variáveis de entrada, o número de prótons Z e o número de massa A , com a condição de que $Z \geq 20$ e $A \geq 40$. A rede é supervisionada e usa 722 núcleos para dados de treino e 98 núcleos como validação. Resultados para o desvio quadrático médio do raio de carga são de cerca de 0.02 fm para os 820 núcleos utilizados. A figura A.4 mostra a diferença entre o raio de carga calculado teoricamente e o determinado experimentalmente para isótopos do chumbo, usando diferentes neurais e modelos teóricos. Mais detalhes podem ser encontrados na Ref. [84].

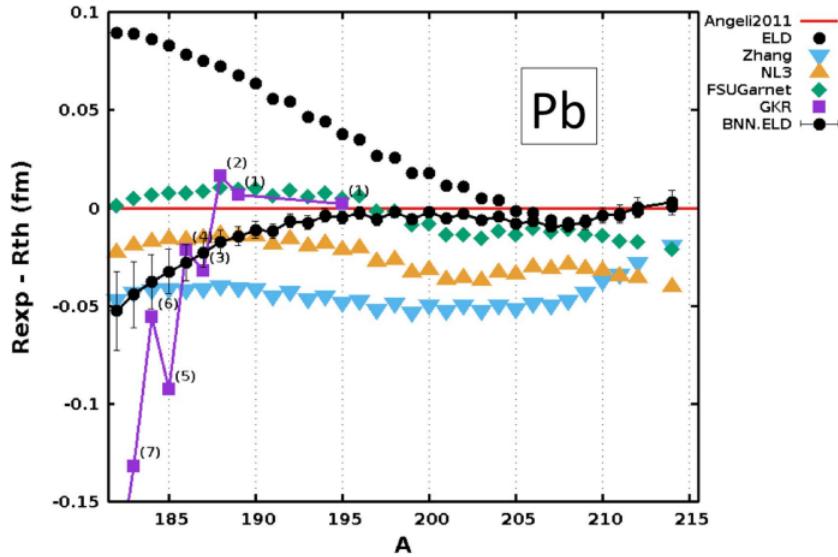


Figura A.4: Previsões para o raio de carga para isótopos do chumbo ($Z = 82$) para diferentes modelos teóricos ou que usam redes neurais. A previsão que contém barras de erro é a que foi brevemente descrita no texto.

A.3 Decaimento β e processo r

O decaimento β é fundamental para entender a origem dos elementos pesados. Prever o tempo de meia vida do decaimento β é de grande importância para simulações do processo r (captura rápida de nêutrons). Com redes neurais é possível fazer previsões que levam em conta a física do problema, como visto na Ref. [85]. O modelo de inteligência artificial leva em conta a teoria de Fermi para o decaimento beta, onde a função f que é minimizada (função custo) é dada por

$$f = \log_{10}(T_{1/2}^a/T_{1/2}^b), \quad (\text{A.1})$$

onde $T_{1/2}^a$ é o tempo de meia vida do decaimento beta medido experimentalmente e

$T_{1/2}^b$ é o tempo de meia vida do decaimento beta determinado teoricamente. As variáveis de entrada são o número de prótons Z , o número de nêtrons N , a energia total do decaimento beta e o parâmetro de paridade $\delta = 1, 0, -1$, para núcleos par-par, ímpar-par e ímpar-ímpar, respectivamente.

A figura A.5 mostra a previsão do tempo de meia vida ($T_{1/2}$) do decaimento β (em segundos) para isótonos com número de nêtrons $N = 126$ usando redes neurais.

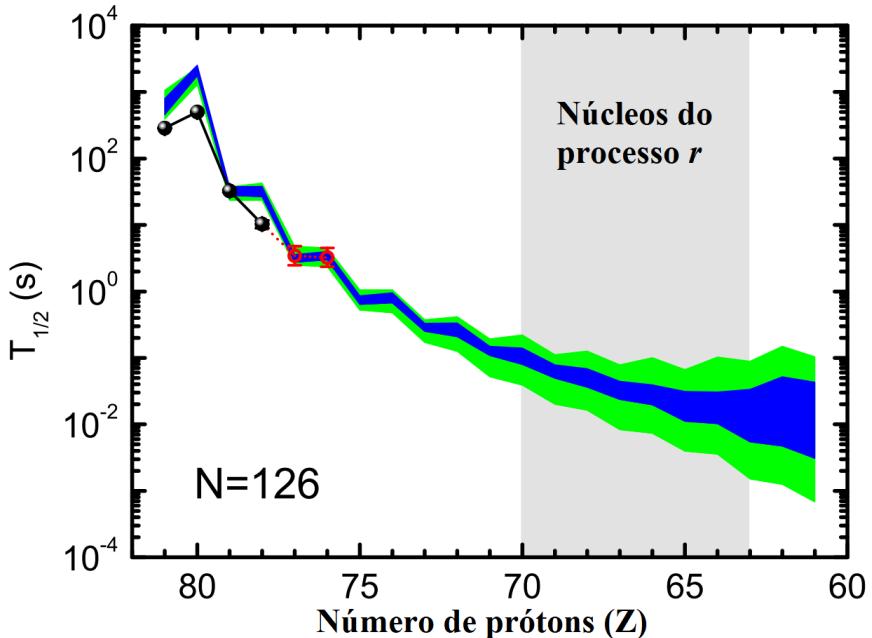


Figura A.5: Meias-vidas de decaimento β para isótonos com $N = 126$. A região hachurada verde mostra as previsões de uma rede neural. A região hachurada em azul mostra os resultados da mesma rede neural, porém seus dados de aprendizado são estendidos para incluir três meias-vidas extras de decaimento β para cada isótopo (indicado por círculos abertos) em direção à *drip-line* de nêtrons [85].

A.4 Alvos ativos

Experimentos com alvos ativos geram enormes quantidades de dados. Uma semana de experimento pode gerar até 10 Tb de dados [86], o que gera a necessidade do uso de algoritmos de machine learning para diminuir o consumo de tempo da análise desses dados.

A análise dos dados envolve a reconstrução tridimensional dos eventos e posteriormente a reconstrução da cinemática, com a identificação de partículas e de reações nucleares. O uso de técnicas de machine learning pode diminuir o consumo de tempo dessas etapas.

O uso de CNNs em imagens feitas a partir das projeções de eventos pode ser útil para a classificação de eventos, sem a necessidade de reconstruir a cinemática em uma etapa anterior. A figura A.6 mostra exemplos de projeções de trajetórias de partículas dentro do alvo ativo, onde a partir da projeção no plano xy é possível identificar a partícula que a originou.

Primeiro, com o objetivo de classificar as projeções entre próton ou carbono, é possível construir uma rede neural para a classificação binária. Para isso foi usada uma arquitetura com camadas convolucionais seguidas de max-pooling e por fim uma camada FC [86]. A função a ser minimizada é a dada pela equação 3.5 e a métrica para entender o comportamento da rede neural é a acurácia binária.

Caso o objetivo seja classificar entre três ou mais possibilidades (como por exemplo próton, carbono e outros), então a classificação não será binária, passará a ser categórica, pois será necessário classificar a imagem em alguma categoria. A função custo passará a ser a *categorical cross-entropy* e a métrica será a acurácia categórica.

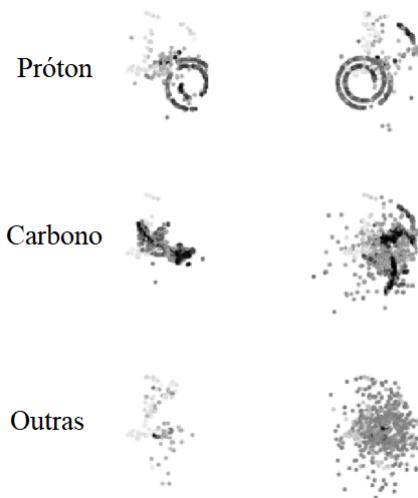


Figura A.6: Projeções no plano xy de partículas dentro do alvo ativo, onde quanto mais escura for a cor, mais carga tem o ponto, e quanto mais clara a cor, menos carga tem o ponto. O objetivo da rede neural pode ser classificar corretamente se a imagem à direita corresponde à uma trajetória de um próton, carbono ou outra partícula, ou pode ser uma rede neural para classificação binária caso seja necessário classificar apenas entre próton ou carbono [86].

A rede criada para a classificação binária (entre próton e carbono) atingiu cerca de 90% de acurácia. Somado ao fato de que redes neurais são muito eficientes em tempo, isso mostra uma possibilidade de que seja possível utilizar redes neurais como essa para o processamento em tempo real dos dados, com pouca supervisão humana durante o processo [86].

Outro problema que pode ser resolvido com técnicas de machine learning é a remoção de ruído de eventos. Para isso, pode ser usado o algoritmo não supervisionado *Hough transform* [87], que é capaz de identificar padrões de interesse nos dados e evitar regiões ou pontos que não seguem o padrão desejado.

Apêndice B

Algoritmo p-RANSAC

Algoritmo 1: p-RANSAC

Dados: pointcloud, N , d_{min} , N_{min}

- 1 **para** cada iteração $i = 1, 2, \dots, N$ **faz**
 - 2 Seleciona dois pontos da nuvem de pontos de modo aleatório (Random Sampling);
 - 3 Estima versor v e um ponto P_b que passe pela reta r formada pelos dois pontos;
 - 4 **para** cada ponto P **faz**
 - 5 Calcula a distância d do ponto à reta r ;
 - 6 **se** $d < d_{min}$ **então**
 - 7 Guarda P como pertencente à r ;
 - 8 **se** Número de pontos de $r > N_{min}$ **então**
 - 9 Guarda v , P_b e C ;
 - 10 Ordena as retas do menor para o maior C ;
 - 11 **para** cada reta r ordenada **faz**
 - 12 **se** Número de pontos de $r > N_{min}$ **então**
 - 13 Guarda v , P_b e pontos $P \in r$;
 - 14 **retorna** Retas r selecionadas na última etapa;
-

Referências

- [1] D.W. Bardayan et al. “Determining the $^{14}\text{O}(\alpha,\text{p})^{17}\text{F}$ astrophysical rate from Measurements at TwinSol”. In: *Physics Procedia* 90 (2017). Conference on the Application of Accelerators in Research and Industry, CAARI 2016, 30 October – 4 November 2016, Ft. Worth, TX, USA, pp. 415–420. ISSN: 1875-3892. DOI: <https://doi.org/10.1016/j.phpro.2017.09.043>. URL: <https://www.sciencedirect.com/science/article/pii/S187538921730202X>.
- [2] Faïrouz Hammache and Nicolas de Séréville. “Transfer Reactions As a Tool in Nuclear Astrophysics”. In: *Frontiers in Physics* 8 (2021). ISSN: 2296-424X. DOI: 10.3389/fphy.2020.602920. URL: <https://www.frontiersin.org/articles/10.3389/fphy.2020.602920>.
- [3] *The colourful nuclide chart beta*. URL: <https://people.physics.anu.edu.au/~ecs103/chart-beta/>.
- [4] J. J. Kolata, V. Guimaraes, and E. F. Aguilera. “Elastic scattering, fusion, and breakup of light exotic nuclei”. In: *The European Physical Journal A* 52.5 (2016), p. 123. ISSN: 1434-601X. DOI: 10.1140/epja/i2016-16123-1. URL: <https://doi.org/10.1140/epja/i2016-16123-1>.
- [5] Y. Kucuk and A. M. Moro. “Exclusive breakup of ^{17}F on ^{58}Ni and ^{208}Pb within the continuum-discretized coupled-channels method”. In: *Phys. Rev. C* 86 (3 2012), p. 034601. DOI: 10.1103/PhysRevC.86.034601. URL: <https://link.aps.org/doi/10.1103/PhysRevC.86.034601>.
- [6] G.F. Fortino et al. “Digital signal analysis based on convolutional neural networks for active target time projection chambers”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 1031 (2022), p. 166497. ISSN: 0168-9002. DOI: <https://doi.org/10.1016/j.nima.2022.166497>. URL: <https://www.sciencedirect.com/science/article/pii/S016890022200119X>.

- [7] J. W. Bradt. “Measurement of isobaric analogue resonances of ^{47}Ar with the Active-Target Time Projection Chamber”. PhD thesis. Michigan State University, USA, 2017. URL: https://publications.nscl.msu.edu/thesis/%20Brandt_2017_5279.pdf.
- [8] J. Bradt et al. “Commissioning of the Active-Target Time Projection Chamber”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 875 (2017), pp. 65–79. ISSN: 0168-9002. DOI: <https://doi.org/10.1016/j.nima.2017.09.013>. URL: <http://www.sciencedirect.com/science/article/pii/S0168900217309683>.
- [9] T. Furuno et al. “Performance test of the MAIKo active target”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 908 (2018), pp. 215–224. ISSN: 0168-9002. DOI: <https://doi.org/10.1016/j.nima.2018.08.042>. URL: <https://doi.org/10.1016/j.nima.2018.08.042>.
- [10] E. Koshchiiy et al. “Texas Active Target (TexAT) detector for experiments with rare isotope beams”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 957 (2020), p. 163398. ISSN: 0168-9002. DOI: <https://doi.org/10.1016/j.nima.2020.163398>. URL: <https://doi.org/10.1016/j.nima.2020.163398>.
- [11] Y. Giomataris et al. “MICROMEGAS: a high-granularity position-sensitive gaseous detector for high particle-flux environments”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 376.1 (1996), pp. 29–35. ISSN: 0168-9002. DOI: [https://doi.org/10.1016/0168-9002\(96\)00175-1](https://doi.org/10.1016/0168-9002(96)00175-1). URL: [https://doi.org/10.1016/0168-9002\(96\)00175-1](https://doi.org/10.1016/0168-9002(96)00175-1).
- [12] J. Giovinazzo et al. “GET electronics samples data analysis”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 840 (2016), pp. 15–27. ISSN: 0168-9002. DOI: <https://doi.org/10.1016/j.nima.2016.09.018>. URL: <https://doi.org/10.1016/j.nima.2016.09.018>.

- [13] D. Suzuki et al. “Prototype AT-TPC: Toward a new generation active target time projection chamber for radioactive beam experiments”. In: *Nuclear Instruments and Methods in Physics Research Section A Accelerators Spectrometers Detectors and Associated Equipment* 691 (Nov. 2012), p. 39. DOI: 10.1016/j.nima.2012.06.050.
- [14] J.C. Zamora and G.F. Fortino. “Tracking algorithms for TPCs using consensus-based robust estimators”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* (2020), p. 164899. ISSN: 0168-9002. DOI: <https://doi.org/10.1016/j.nima.2020.164899>. URL: <http://www.sciencedirect.com/science/article/pii/S0168900220312961>.
- [15] F.D. Becchetti et al. “The TwinSol low-energy radioactive nuclear beam apparatus: status and recent results”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 505.1 (2003). Proceedings of the tenth Symposium on Radiation Measurements and Applications, pp. 377 –380. ISSN: 0168-9002. DOI: [https://doi.org/10.1016/S0168-9002\(03\)01101-X](https://doi.org/10.1016/S0168-9002(03)01101-X). URL: <http://www.sciencedirect.com/science/article/pii/S016890020301101X>.
- [16] J.J. Kolata et al. “A radioactive beam facility using a large superconducting solenoid”. In: *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms* 40-41 (1989), pp. 503–506. ISSN: 0168-583X. DOI: [https://doi.org/10.1016/0168-583X\(89\)91032-X](https://doi.org/10.1016/0168-583X(89)91032-X). URL: <https://www.sciencedirect.com/science/article/pii/0168583X8991032X>.
- [17] Panagiotis Gastis et al. “Improved phenomenological description of equilibrium charge state distributions for Ni, Co, and Cu ions in Mo based on new experimental data at 2 MeV/u”. In: (Sept. 2015).
- [18] L. E. Tamayose et al. “Simulation of the RIBRAS Facility with GEANT4”. In: *Brazilian Journal of Physics* 52.3 (2022), p. 89. ISSN: 1678-4448. DOI: 10.1007/s13538-022-01090-y. URL: <https://doi.org/10.1007/s13538-022-01090-y>.
- [19] J.C. Zamora. “Estudo do espalhamento elástico dos isótopos 7Be, 9Be e 10Be em alvo de 12C”. MA thesis. University of Sao Paulo, Brazil, 2011. DOI: 10.11606/D.43.2011.tde-30092011-132427.

- [20] S. Agostinelli et al. “Geant4—a simulation toolkit”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 506.3 (2003), pp. 250–303. ISSN: 0168-9002. DOI: [https://doi.org/10.1016/S0168-9002\(03\)01368-8](https://doi.org/10.1016/S0168-9002(03)01368-8). URL: <https://www.sciencedirect.com/science/article/pii/S0168900203013688>.
- [21] R. Lichtenthäler et al. “Radioactive ion beams in Brasil (RIBRAS)”. en. In: *Brazilian Journal of Physics* 33 (June 2003), pp. 294 –296. ISSN: 0103-9733. URL: http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0103-97332003000200025&nrm=iso.
- [22] Sérgio R. Muniz, Vanderlei S. Bagnato, and M. Bhattacharya. “Analysis of off-axis solenoid fields using the magnetic scalar potential: An application to a Zeeman-slower for cold atoms”. In: *American Journal of Physics* 83.6 (2015), pp. 513–517. DOI: 10.1119/1.4906516. URL: <https://doi.org/10.1119%2F1.4906516>.
- [23] B. Mauss et al. “Commissioning of the ACTive TARget and Time Projection Chamber (ACTAR TPC)”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 940 (2019), pp. 498–504. ISSN: 0168-9002. DOI: <https://doi.org/10.1016/j.nima.2019.06.067>. URL: <https://www.sciencedirect.com/science/article/pii/S016890021930912X>.
- [24] J.S. Randhawa et al. “Beam-induced space-charge effects in time projection chambers in low-energy nuclear physics experiments”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 948 (2019), p. 162830. ISSN: 0168-9002. DOI: <https://doi.org/10.1016/j.nima.2019.162830>. URL: <https://www.sciencedirect.com/science/article/pii/S0168900219312641>.
- [25] Thomas Lohse and Werner Witzeling. “The Time Projection Chamber”. In: *Instrumentation in High Energy Physics*, pp. 81–155. DOI: 10.1142/9789814360333_0002. eprint: https://www.worldscientific.com/doi/pdf/10.1142/9789814360333_0002. URL: https://www.worldscientific.com/doi/abs/10.1142/9789814360333_0002.

- [26] A. Geron. *Hands-on machine learning with Scikit-Learn, Keras and TensorFlow: concepts, tools, and techniques to build intelligent systems*. 2nd ed. O'Reilly, 2019. ISBN: 9781492032649.
- [27] *Machine Learning and Data Analysis for Nuclear Physics*. 2020. URL: <https://ectstar.fbk.eu/node/4472>.
- [28] Maryam M. Najafabadi et al. “Deep learning applications and challenges in big data analytics”. In: *Journal of Big Data* 2.1 (2015), p. 1. ISSN: 2196-1115. DOI: 10.1186/s40537-014-0007-7. URL: <https://doi.org/10.1186/s40537-014-0007-7>.
- [29] Morten Hjorth-Jensen. *Data Analysis and Machine Learning: Neural networks, from the simple perceptron to deep learning*. 2020. URL: <https://nucleartalent.github.io/MachineLearningECT/doc/pub/Day4/pdf/Day4.pdf> (visited on 08/2020).
- [30] C.M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer New York, 2016. ISBN: 9781493938438. URL: <https://books.google.com.br/books?id=kOXDtAEACAAJ>.
- [31] Steven B. Damelin and Willard Miller Jr. *The Mathematics of Signal Processing*. Cambridge Texts in Applied Mathematics. Cambridge University Press, 2011. DOI: 10.1017/CBO9781139003896.
- [32] K.-L Du and M.N.S Swamy. “Radial Basis Function Networks”. In: Dec. 2014, pp. 299–335. ISBN: 978-1-4471-5570-6. DOI: 10.1007/978-1-4471-5571-3_10.
- [33] Alex Sherstinsky. “Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network”. In: *Physica D: Nonlinear Phenomena* 404 (2020), p. 132306. ISSN: 0167-2789. DOI: <https://doi.org/10.1016/j.physd.2019.132306>. URL: <https://www.sciencedirect.com/science/article/pii/S0167278919305974>.
- [34] Abien Fred Agarap. “Deep Learning using Rectified Linear Units (ReLU)”. In: *arXiv e-prints*, arXiv:1803.08375 (2018), arXiv:1803.08375. arXiv: 1803.08375 [cs.NE].
- [35] Moraga C. Han J. “The influence of the sigmoid function parameters on the speed of backpropagation learning”. In: *Lecture Notes in Computer Science* 930 (1995). DOI: https://doi.org/10.1007/3-540-59497-3_175.

- [36] Tomasz Szandała. *Review and Comparison of Commonly Used Activation Functions for Deep Neural Networks*. Oct. 2020. DOI: https://doi.org/10.1007/978-981-15-5495-7_11.
- [37] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. <http://www.deeplearningbook.org>. MIT Press, 2016.
- [38] Hecht-Nielsen. “Theory of the backpropagation neural network”. In: *International 1989 Joint Conference on Neural Networks*. 1989, 593–605 vol.1. DOI: [10.1109/IJCNN.1989.118638](https://doi.org/10.1109/IJCNN.1989.118638).
- [39] Douglas Kline and Victor Berardi. “Revisiting squared-error and cross-entropy functions for training neural network classifiers”. In: *Neural Computing and Applications* 14 (Dec. 2005), pp. 310–318. DOI: [10.1007/s00521-005-0467-y](https://doi.org/10.1007/s00521-005-0467-y).
- [40] Diederik P. Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In: (2014). DOI: [10.48550/ARXIV.1412.6980](https://doi.org/10.48550/ARXIV.1412.6980). URL: <https://arxiv.org/abs/1412.6980>.
- [41] Sebastian Ruder. “An overview of gradient descent optimization algorithms”. In: *ArXiv abs/1609.04747* (2016).
- [42] Mohammad Hossin and Sulaiman M.N. “A Review on Evaluation Metrics for Data Classification Evaluations”. In: *International Journal of Data Mining & Knowledge Management Process* 5 (Mar. 2015), pp. 01–11. DOI: [10.5121/ijdkp.2015.5201](https://doi.org/10.5121/ijdkp.2015.5201).
- [43] N. S. Altman. “An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression”. In: *The American Statistician* 46.3 (1992), pp. 175–185. DOI: [10.1080/00031305.1992.10475879](https://doi.org/10.1080/00031305.1992.10475879). eprint: <https://doi.org/10.1080/00031305.1992.10475879>. URL: <https://www.tandfonline.com/doi/abs/10.1080/00031305.1992.10475879>.
- [44] Angela Serra and Roberto Tagliaferri. “Unsupervised Learning: Clustering”. In: Jan. 2018. ISBN: 9780128096338. DOI: [10.1016/B978-0-12-809633-8.20487-1](https://doi.org/10.1016/B978-0-12-809633-8.20487-1).
- [45] Kristina Sinaga and Miin-Shen Yang. “Unsupervised K-Means Clustering Algorithm”. In: *IEEE Access* PP (Apr. 2020), pp. 1–1. DOI: [10.1109/ACCESS.2020.2988796](https://doi.org/10.1109/ACCESS.2020.2988796).

- [46] Niago Moreira Nobre Leite et al. “Deep Convolutional Autoencoder for EEG Noise Filtering”. In: *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. 2018, pp. 2605–2612. DOI: [10.1109/BIBM.2018.8621080](https://doi.org/10.1109/BIBM.2018.8621080).
- [47] Y Reddy, Viswanath Pulabaigari, and Eswara B. “Semi-supervised learning: a brief review”. In: *International Journal of Engineering & Technology* 7 (Feb. 2018), p. 81. DOI: [10.14419/ijet.v7i1.8.9977](https://doi.org/10.14419/ijet.v7i1.8.9977).
- [48] Michael L Littman Leslie Pack Kaelbling and Andrew W Moore. “Reinforcement learning: A survey”. In: *Journal of artificial intelligence research* (1996), 237–285. DOI: <https://doi.org/10.1613/jair.301>.
- [49] Kim D et al. “Review of machine learning methods in soft robotics”. In: *PLoS ONE* (2021). DOI: <https://doi.org/10.1371/journal.pone.0246102>.
- [50] P. Holl et al. “Deep learning based pulse shape discrimination for germanium detectors”. In: *The European Physical Journal C* 79.6 (2019), p. 450. ISSN: 1434-6052. DOI: [10.1140/epjc/s10052-019-6869-2](https://doi.org/10.1140/epjc/s10052-019-6869-2). URL: <https://doi.org/10.1140/epjc/s10052-019-6869-2>.
- [51] Edmundo Capelas de Oliveira and José Emilío Maiorino. *Introdução aos métodos da matemática aplicada*. 3rd ed. Editora UNICAMP, 2010. ISBN: 9788526809062.
- [52] *ROOT Data Analysis Framework*. 2021. URL: <https://root.cern.ch/>.
- [53] C.G. Ryan et al. “SNIP, a statistics-sensitive background treatment for the quantitative analysis of PIXE spectra in geoscience applications”. In: *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms* 34.3 (1988), pp. 396–402. ISSN: 0168-583X. DOI: [https://doi.org/10.1016/0168-583X\(88\)90063-8](https://doi.org/10.1016/0168-583X(88)90063-8). URL: <https://www.sciencedirect.com/science/article/pii/0168583X88900638>.
- [54] Miroslav Morháč et al. “Background elimination methods for multidimensional coincidence γ -ray spectra”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 401.1 (1997), pp. 113–132. ISSN: 0168-9002. DOI: [https://doi.org/10.1016/S0168-9002\(97\)01023-1](https://doi.org/10.1016/S0168-9002(97)01023-1). URL: <https://www.sciencedirect.com/science/article/pii/S0168900297010231>.
- [55] Donald D. Burgess and Richard J. Tervo. “Background estimation for gamma-ray spectrometry”. In: *Nuclear Instruments and Methods in Physics Research* 214.2 (1983), pp. 431–434. ISSN: 0167-5087. DOI: <https://doi.org/10.1016/>

- 0167-5087 (83) 90612-9. URL: <https://www.sciencedirect.com/science/article/pii/0167508783906129>.
- [56] Miroslav Morháč et al. “Identification of peaks in multidimensional coincidence γ -ray spectra”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 443.1 (2000), pp. 108–125. ISSN: 0168-9002. DOI: [https://doi.org/10.1016/S0168-9002\(99\)01005-0](https://doi.org/10.1016/S0168-9002(99)01005-0). URL: <https://www.sciencedirect.com/science/article/pii/S0168900299010050>.
- [57] Pauli Virtanen et al. “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python”. In: *Nature Methods* 17 (2020), pp. 261–272. DOI: [10.1038/s41592-019-0686-2](https://doi.org/10.1038/s41592-019-0686-2).
- [58] Matt Newville et al. *lmfit/lmfit-py 1.0.2*. Version 1.0.2. Feb. 2021. DOI: [10.5281/zenodo.4516651](https://doi.org/10.5281/zenodo.4516651). URL: <https://doi.org/10.5281/zenodo.4516651>.
- [59] Xavier Glorot and Y. Bengio. “Understanding the difficulty of training deep feed-forward neural networks”. In: *Journal of Machine Learning Research - Proceedings Track 9* (Jan. 2010), pp. 249–256.
- [60] Ekaba Bisong. “Google Colaboratory”. In: *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners*. Berkeley, CA: Apress, 2019, pp. 59–64. ISBN: 978-1-4842-4470-8. DOI: [10.1007/978-1-4842-4470-8_7](https://doi.org/10.1007/978-1-4842-4470-8_7). URL: https://doi.org/10.1007/978-1-4842-4470-8_7.
- [61] Andrea Dal Pozzolo et al. “Calibrating Probability with Undersampling for Unbalanced Classification”. In: Dec. 2015. DOI: [10.1109/SSCI.2015.33](https://doi.org/10.1109/SSCI.2015.33).
- [62] Ashraf A Aly, Safaai Bin Deris, and Nazar Zaki. “Research review for digital image segmentation techniques”. In: *Int. J. Comput. Sci. Inf. Technol. Res.* 3.5 (2011), p. 99.
- [63] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. “Open3D: A Modern Library for 3D Data Processing”. In: *arXiv:1801.09847* (2018).
- [64] Erich Schubert et al. “DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN”. In: *ACM Trans. Database Syst.* 42.3 (2017). ISSN: 0362-5915. DOI: [10.1145/3068335](https://doi.org/10.1145/3068335). URL: <https://doi.org/10.1145/3068335>.

- [65] Leland McInnes, John Healy, and Steve Astels. “hdbscan: Hierarchical density based clustering”. In: *The Journal of Open Source Software* 2.11 (2017), p. 205.
- [66] Leland McInnes and John Healy. “Accelerated Hierarchical Density Based Clustering”. In: *Data Mining Workshops (ICDMW), 2017 IEEE International Conference on*. IEEE. 2017, pp. 33–42.
- [67] Sam Fletcher and Md Islam. “Comparing sets of patterns with the Jaccard index”. In: *Australasian Journal of Information Systems* 22 (Mar. 2018). DOI: 10.3127/ajis.v22i0.1538.
- [68] Peter J. Rousseeuw. “Silhouettes: A graphical aid to the interpretation and validation of cluster analysis”. In: *Journal of Computational and Applied Mathematics* 20 (1987), pp. 53 –65. ISSN: 0377-0427. DOI: [https://doi.org/10.1016/0377-0427\(87\)90125-7](https://doi.org/10.1016/0377-0427(87)90125-7). URL: <http://www.sciencedirect.com/science/article/pii/0377042787901257>.
- [69] Charles R Qi et al. “PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation”. In: *arXiv preprint arXiv:1612.00593* (2016).
- [70] JMaosheng Tang. “Semantic Classification of 3D Point Cloud by Random Forest-based Feature Learning”. MA thesis. Technische Universität Berlin, Germany, 2019.
- [71] Yassid Ayyad et al. “Novel particle tracking algorithm based on the Random Sample Consensus Model for the Active Target Time Projection Chamber (AT-TPC)”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 880 (2018), pp. 166 – 173. ISSN: 0168-9002. DOI: <https://doi.org/10.1016/j.nima.2017.10.090>. URL: <http://www.sciencedirect.com/science/article/pii/S0168900217311798>.
- [72] Charles R Qi et al. “PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space”. In: *arXiv preprint arXiv:1706.02413* (2017).
- [73] Yue Wang et al. *Dynamic Graph CNN for Learning on Point Clouds*. 2018. DOI: 10.48550/ARXIV.1801.07829. URL: <https://arxiv.org/abs/1801.07829>.
- [74] Ling Zhang and Zhigang Zhu. *Unsupervised Feature Learning for Point Cloud by Contrasting and Clustering With Graph Convolutional Neural Network*. 2019. DOI: 10.48550/ARXIV.1904.12359. URL: <https://arxiv.org/abs/1904.12359>.

- [75] MP Kuchera et al. “LISE++ Software Updates and Future Plans”. In: *Journal of Physics: Conference Series* 664.7 (2015), p. 072029. DOI: 10.1088/1742-6596/664/7/072029. URL: <https://doi.org/10.1088/1742-6596/664/7/072029>.
- [76] William R Leo. “Techniques for nuclear and particle physics experiments”. In: *Nucl Instrum Methods Phys Res* 834 (1988), p. 290.
- [77] Mário José de Oliveira. *Termodinâmica*. Livraria da Física, 2005.
- [78] J.F. Liang et al. “Dynamic polarization in the Coulomb breakup of loosely bound 17F”. In: *Physics Letters B* 681.1 (2009), pp. 22–25. ISSN: 0370-2693. DOI: <https://doi.org/10.1016/j.physletb.2009.09.055>. URL: <https://www.sciencedirect.com/science/article/pii/S0370269309011472>.
- [79] C.A Bertulani and P Danielewicz. “Breakup of the weakly bound 17F nucleus”. In: *Nuclear Physics A* 717.3 (2003), pp. 199–213. ISSN: 0375-9474. DOI: [https://doi.org/10.1016/S0375-9474\(03\)00640-7](https://doi.org/10.1016/S0375-9474(03)00640-7). URL: <https://www.sciencedirect.com/science/article/pii/S0375947403006407>.
- [80] L. Yang et al. “Insight into the reaction dynamics of proton drip-line nuclear system 17F+58Ni at near-barrier energies”. In: *Physics Letters B* 813 (2021), p. 136045. ISSN: 0370-2693. DOI: <https://doi.org/10.1016/j.physletb.2020.136045>. URL: <https://www.sciencedirect.com/science/article/pii/S0370269320308480>.
- [81] Stefano Carboni et al. “Particle identification using the (DELTA)E-E technique and pulse shape discrimination with the silicon detectors of the FAZIA project”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 664 (Feb. 2012), 251–263. DOI: 10.1016/j.nima.2011.10.061.
- [82] Guolin Ke et al. “LightGBM: A Highly Efficient Gradient Boosting Decision Tree”. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS’17. Long Beach, California, USA: Curran Associates Inc., 2017, 3149–3157. ISBN: 9781510860964.
- [83] Ze-Peng Gao et al. “Machine learning the nuclear mass”. In: *Nuclear Science and Techniques* 32.10 (2021), p. 109. ISSN: 2210-3147. DOI: 10.1007/s41365-021-00956-1. URL: <https://doi.org/10.1007/s41365-021-00956-1>.

- [84] R Utama, Wei-Chia Chen, and J Piekarewicz. “Nuclear charge radii: density functional theory meets Bayesian neural networks”. In: *Journal of Physics G: Nuclear and Particle Physics* 43.11 (2016), p. 114002. DOI: 10.1088/0954-3899/43/11/114002. URL: <https://doi.org/10.1088/0954-3899/43/11/114002>.
- [85] Z. M. Niu et al. “Predictions of nuclear β -decay half-lives with machine learning and their impact on r-process nucleosynthesis”. In: *Phys. Rev. C* 99 (6 2019), p. 064307. DOI: 10.1103/PhysRevC.99.064307. URL: <https://link.aps.org/doi/10.1103/PhysRevC.99.064307>.
- [86] M.P. Kuchera et al. “Machine learning methods for track classification in the AT-TPC”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 940 (2019), pp. 156–167. ISSN: 0168-9002. DOI: <https://doi.org/10.1016/j.nima.2019.05.097>. URL: <https://www.sciencedirect.com/science/article/pii/S0168900219308046>.
- [87] Christoph Dalitz, Tilman Schramke, and Manuel Jeltsch. “Iterative Hough Transform for Line Detection in 3D Point Clouds”. In: *Image Processing On Line* 7 (2017). <https://doi.org/10.5201/ipol.2017.208>, pp. 184–196.