



CENTRO BRASILEIRO DE PESQUISAS FÍSICAS

# **Eventos de carga pontual em CONNIE: simulações e análise com redes neurais convolucionais**

**Ana Carolina Oliveira da Silva Sá**

Trabalho apresentado ao Programa  
de Pós-Graduação no Centro Brasi-  
leiro de Pesquisas Físicas como requi-  
sito parcial para obtenção do grau de  
Mestre em Física.

**Orientadora: Dra. Carla Bonifazi**  
**Coorientador: Dr. Martín Makler**

Rio de Janeiro, RJ  
2025

Ana Carolina Oliveira da Silva Sá

**Eventos de carga pontual em CONNIE:  
simulações e análise com redes neurais convolucionais**

Trabalho apresentado ao Programa de Pós-Graduação no Centro Brasileiro de Pesquisas Físicas como requisito parcial para obtenção do grau de Mestre em Física.

CENTRO BRASILEIRO DE PESQUISAS FÍSICAS

Orientador: Dra. Carla Bonifazi  
Coorientador: Dr. Martín Makler


Rio de Janeiro, RJ  
2025




"EVENTOS DE CARGA PONTUAL EM CONNIE: SIMULAÇÃO E ANÁLISE COM  
REDES NEURAIS CONVOLUCIONAIS"

**ANA CAROLINA OLIVEIRA DA SILVA SÁ**

Dissertação de Mestrado em Física apresentada no  
Centro Brasileiro de Pesquisas Físicas do  
Ministério da Ciência Tecnologia e Inovação.  
Fazendo parte da banca examinadora os seguintes  
professores:



Carla Brenda Bonifazi - Orientadora/CBPF



Martin Makler – Coorientador/ CBPF



Hiroshi Nunokawa – PUC/RJ



Edivaldo Moura Santos – USP

Rio de Janeiro, 09 de outubro de 2025.

---

da Silva Sá, Ana Carolina Oliveira

**Eventos de carga pontual em CONNIE:**

**simulações e análise com redes neurais convolucionais / Ana Carolina Oliveira**

**da Silva Sá.** - 2025

139 f. : il.

– CENTRO BRASILEIRO DE PESQUISAS FÍSICAS, Rio de Janeiro, RJ, 2025.

Orientador: Dra. Carla Bonifazi

1. Neutrinos de Reator. 2. Skipper-CCDs. 3. Rede Neural Convolucional.

I. Eventos de carga pontual em CONNIE: simulações e análise com redes neurais convolucionais.

---



# Agradecimentos

Agradeço a Deus, por me conduzir ao longo de todo este percurso, pela dádiva da vida, pelo amparo nos momentos difíceis e pela alegria da Sua companhia.

Aos meus pais, Ana Paula e Cláudio, por sempre acreditarem que eu poderia ir além, pelo esforço dedicado nas longas horas de trabalho para que eu pudesse estudar e realizar meus sonhos, e por serem o porto seguro para onde sempre posso voltar.

Aos meus avós, Maria e Paulo, que tornam qualquer ambiente mais acolhedor, pelo cuidado e carinho sempre presentes.

Aos meus irmãos, Erick Jhonson e Daniel, pelo companheirismo, apoio e referência que me dão todos os dias. Tenho muito orgulho de como construímos nossas histórias e de como superamos as limitações que a vida nos impôs.

Ao meu noivo, Gabriel, pelo amor, incentivo e cuidado demonstrados em cada detalhe e que tornam tudo mais leve e bonito.

Ao meu coorientador, Prof. Dr. Martin Makler, por ter aceitado colaborar com este trabalho. Pela paciência e atenção dedicados e pelas contribuições fundamentais que tornaram tudo isso possível.

À Prof. Dra. Carla Bonifazi, minha orientadora desde a graduação, que me apresentou a física experimental, pelo apoio constante e pela inspiração como pesquisadora. No início da minha trajetória acadêmica as muitas dificuldades que enfrentei me fizeram duvidar se um dia eu poderia participar de um experimento ou desenvolver pesquisas... se um dia eu seria uma física. Hoje, isso é realidade, e eu sou grata a você, pela oportunidade de enxergar que este lugar também poderia ser meu. É uma grande honra ser sua aluna.

Aos amigos que fiz durante a caminhada, Katherine, Pedro, Patrick, Isabelle, Carolina, Thaís, Gabriela, Stalin e Jéssica, que tornaram tudo mais colorido e divertido, por sempre me incentivarem a continuar.

À Colaboração CONNIE, por me receberem e pela oportunidade de aprender e crescer como cientista.

À CAPES, pelo suporte financeiro que permitiu este trabalho.

# Resumo

O experimento CONNIE (Coherent Neutrino Nucleus Interaction Experiment) opera próximo ao reator nuclear de Angra 2 e tem como principal objetivo detectar, pela primeira vez em silício, o espalhamento elástico coerente de antineutrinos com núcleos ( $\text{CE}\nu\text{NS}$ ). Esse tipo de interação produz sinais mensuráveis da ordem de poucos keV, o que requer tecnologias com baixíssimo limiar de detecção. Em meados de 2021, CONNIE passou a utilizar uma tecnologia inovadora de detecção, os Skipper-CCDs. Estes sensores semicondutores pixelados se destacam por seu processo de leitura sequencial e não destrutivo, que permite múltiplas medições das cargas acumuladas em cada pixel. Devido a este mecanismo, os Skipper-CCDs alcançam ruídos de leitura subeletrônicos, o que os torna especialmente apropriados para detectar interações fracas, como  $\text{CE}\nu\text{NS}$ . Neste contexto de altíssima sensibilidade, um dos maiores desafios está na rejeição de ruído de fundo, para a identificação eficiente dos eventos candidatos a  $\text{CE}\nu\text{NS}$ . Uma forma de diferenciar estes eventos de outras interações que geram sinais similares de carga pontual, como raios X e fótons de baixas energias, é através da profundidade onde a ionização foi produzida. Atualmente, o algoritmo que o experimento utiliza para estimar essa profundidade é baseado em ajustes gaussianos, que são imprecisos para reconstruir profundidades de eventos gerados próximos à superfície dos sensores. Neste trabalho, desenvolvemos e otimizamos redes neurais convolucionais para estimar a profundidade de eventos de carga pontual em imagens de Skipper-CCD. Para possibilitar o treinamento e a validação destas redes, implementamos e validamos um código de simulação capaz de gerar imagens com eventos de diferentes energias e profundidades. Este código evoluiu para uma ferramenta útil em estudos complementares a este trabalho, permitindo investigar de forma mais ampla a resposta do sensor em diferentes condições de operação. A arquitetura mais adequada para o problema foi escolhida através de um estudo extensivo de hiperparâmetros. A performance obtida pela rede neural selecionada supera a do algoritmo usual de CONNIE, com previsões mais acuradas e precisas em grandes profundidades.

**Palavras Chaves:** Neutrinos de Reator; Skipper-CCDs; Rede Neural Convolucional.

# Abstract

The CONNIE experiment (Coherent Neutrino Nucleus Interaction Experiment) operates near the Angra 2 nuclear reactor and aims to achieve the first-ever detection of coherent elastic scattering of antineutrinos off nuclei ( $\text{CE}\nu\text{NS}$ ) in silicon. This type of interaction produces measurable signals on the order of a few keV, which requires technologies with extremely low detection thresholds. In mid-2021, CONNIE began using an innovative detection technology, the Skipper-CCDs. These pixelated semiconductor sensors stand out for their sequential and non-destructive readout process, which allows multiple measurements of the charge accumulated in each pixel. Due to this mechanism, Skipper-CCDs achieve sub-electron readout noise, making them especially suitable for detecting weak interactions such as  $\text{CE}\nu\text{NS}$ . In this context of extremely high sensitivity, one of the greatest challenges lies in rejecting background noise to efficiently identify  $\text{CE}\nu\text{NS}$  candidate events. One way to distinguish these events from other interactions that generate similar point-charge signals, such as X-rays and low-energy photons, is through the depth at which ionization occurs. Currently, the algorithm used by the experiment to estimate this depth is based on Gaussian fits, which prove inadequate for events generated near the sensor. In this work, we developed and optimized convolutional neural networks to estimate the depth of point-charge events in Skipper-CCD images. To enable the training and validation of these networks, we implemented and validated a simulation code capable of generating images with events of different energies and depths. This code evolved into a useful tool for complementary studies, allowing a broader investigation of the sensor response under different operating conditions. The most suitable architecture for the problem was selected through an extensive hyperparameter study. The performance of the chosen neural network surpasses that of the standard CONNIE algorithm, providing more accurate and precise predictions at large depths.

**Key-Words:** Reactor Neutrinos; Skipper-CCDs; Convolutional Neural Network

# Sumário

Sumário	9	
Lista de ilustrações	11	
Lista de tabelas	18	
1	INTRODUÇÃO	19
2	DISPOSITIVOS DE CARGA ACOPLADA	25
2.1	Criação e Armazenamento das Cargas	26
2.2	Processo de Leitura Sequencial	27
2.3	Imagens de um CCD e extração de eventos	29
2.4	Skipper-CCDs	31
2.5	Difusão das Cargas	33
3	EXPERIMENTO CONNIE	39
3.1	Arranjo Experimental	40
3.1.1	CCDs	42
3.1.2	Skipper-CCDs	42
3.1.3	Multi-Chip-Module (MCM)	44
3.2	Processamento de Dados	44
3.3	Calibração das Imagens	46
3.4	Identificação de Eventos e Análise de Dados	48
3.5	Resultados Científicos	50
4	SIMULAÇÃO DE IMAGENS	55
4.1	Simulação de imagens vazias	55
4.1.1	Ruído de leitura	55
4.1.2	Eventos de Elétron Único	57
4.1.3	Somando as distribuições de fonte de ruído associadas ao pixel	58
4.2	Simulação dos eventos de carga pontual	60
4.3	Validação do algoritmo de Simulação	62
4.3.1	Caracterização das Simulações	62
4.3.2	Comparação de Espectros	64
5	REDES NEURAIAS CONVOLUCIONAIS	67
5.1	Introdução	67



<b>5.2</b>	<b>Redes Neurais Convolucionais</b>	<b>71</b>
5.2.1	Filtros Convolucionais	72
5.2.2	Camadas Convolucionais	73
5.2.3	Camadas de Ativação	75
5.2.4	Camadas de Agrupamento	76
<b>5.3</b>	<b>Aprendizagem da rede neural</b>	<b>78</b>
5.3.1	Funções de Perda	78
5.3.2	Retropropagação	79
5.3.3	Algoritmos de minimização da função de perda	81
<b>5.4</b>	<b>Treinamento, Validação e Teste</b>	<b>82</b>
5.4.1	<i>Cross Validation</i>	84
5.4.2	Técnicas de Regularização	84
<b>6</b>	<b>RESULTADOS</b>	<b>87</b>
<b>6.1</b>	<b><i>Dataset</i></b>	<b>87</b>
<b>6.2</b>	<b>Pré Processamento</b>	<b>88</b>
<b>6.3</b>	<b>Busca de uma CNN adequada para o problema</b>	<b>88</b>
6.3.1	Modelos A	89
6.3.2	Modelos B	94
6.3.3	Modelos C	98
6.3.4	K-Fold Cross Validation	102
6.3.5	Modelos D	108
6.3.6	Modelos E	113
6.3.7	Modelos F	118
6.3.8	Modelos G	121
<b>6.4</b>	<b>Comparação do desempenho da rede (Modelo G.2) com o algoritmo usual de CONNIE</b>	<b>125</b>
<b>7</b>	<b>CONSIDERAÇÕES FINAIS</b>	<b>127</b>
<b>A</b>	<b>RECONSTRUÇÃO DE FALSOS POSITIVOS</b>	<b>131</b>

## Lista de ilustrações

Figura 1.1 – Fluxo de antineutrinos do elétron em função da energia calculado para a posição onde o experimento CONNIE está instalado, há 30 metros do reator nuclear da usina de Angra 2. Figura extraída de [21]. . . . .	22
Figura 2.1 – Estrutura de um pixel MOS, formado por uma junção PN, uma camada de óxido e os <i>gates</i> (a). Cargas armazenadas na região de depleção em decorrência da aplicação de uma tensão de polarização nos <i>gates</i> (b). Representações pictóricas de um pixel MOS, com proporções reais não preservadas. . . . .	26
Figura 2.2 – Processo de geração de cargas no volume ativo do detector. Vide detalhes no texto. Fonte: Colaboração CONNIE. . . . .	27
Figura 2.3 – Processo de leitura sequencial das cargas, onde os pacotes de carga dos pixels são movidos linha a linha por meio dos potenciais $P_i$ , e quando no registro horizontal, são movimentados com os potenciais $H_i$ até o amplificador. Figura alterada de [21]. . . . .	28
Figura 2.4 – Imagem FITS com diferentes assinaturas características. Figura extraída de [21]. . . . .	29
Figura 2.5 – Eventos extraídos e catalogados. À esquerda temos um traço de um múon, ao centro um evento de carga pontual e à direita a assinatura geométrica de um elétron. . . . .	30
Figura 2.6 – Esquematização do processo de leitura não destrutiva de um Skipper-CCD. Figura extraída de [32]. . . . .	31
Figura 2.7 – Evolução da resolução de imagens de Skipper-CCDs produzidas com 4 (à esquerda), 40 (ao centro) e 400 (à direita) amostras por pixel. Figura extraída de [21]. . . . .	32
Figura 2.8 – Distribuição de dados de um conjunto de imagens produzidas por Skipper-CCD. As imagens foram somadas para aumentar a estatística de contagem em cada um dos picos. . . . .	33
Figura 2.9 – Os portadores do pacote de cargas são difundidos pelo campo elétrico $E$ até os poços potenciais na superfície dos pixels. A dispersão é proporcional ao tempo $t_0$ . Figura extraída de [33]. . . . .	34

Figura 2.10–Traço de um múon que atravessa toda o volume do CCD (à esquerda) e curva de calibração largura-profundidade (à direita). Figura retirada de [34]. . . . .	34
Figura 2.11–Difusão de um pacote de carga criado em uma profundidade $z$ (à esquerda), que produz o padrão de difusão com largura igual a $\sigma_{xy}$ (à direita). Figura extraída de [35]. . . . .	35
Figura 2.12–Imagem com único evento de carga pontual simulado. Acima temos a projeção da distribuição de dados no eixo $x$ e na lateral direita temos a projeção no eixo $y$ . Um ajuste gaussiano é feito sobre cada projeção a fim de obter o desvio padrão, que representa a largura. . . . .	36
Figura 3.1 – À esquerda está representada a localização do experimento CONNIE, na Central Nuclear Almirante Álvaro Alberto. À direita temos indicado o contêiner, onde o aparato experimental está instalado, posicionado ao lado do domo de contenção do reator de Angra 2. Figura adaptada de <a href="https://www.eletronuclear.gov.br/">https://www.eletronuclear.gov.br/</a> . . . . .	39
Figura 3.2 – Linha do tempo mostrando a evolução das diferentes fases e <i>setups</i> experimentais do CONNIE ao longo da última década. . . . .	40
Figura 3.3 – À esquerda temos o sensor CCD de 16 megapixels montado sobre a moldura de cobre e conectado a um cabo flexível. À direita, a caixa de cobre onde estes sensores são posicionados em pilha. Figuras extraídas de [34]. . . . .	41
Figura 3.4 – Blindagem passiva do experimento composta por duas camadas de polietileno com 30 cm de espessura e uma de chumbo, de 15 cm. Ao centro temos a câmara de vácuo onde os sensores são dispostos. Figura extraída de [34]. . . . .	41
Figura 3.5 – Skipper-CCD de 1022 x 682 pixels montado sobre um substrato de silício e acoplado a um suporte de cobre junto a um cabo flexível. Figura extraída de [37]. . . . .	43
Figura 3.6 – Arranjo de 16 sensores Skipper-CCDs montados sobre um suporte e conectado a um cabo flexível para leitura. Figura extraída de [42]. . . . .	43
Figura 3.7 – Imagem de Skipper-CCD com as três regiões. A região delimitada à esquerda é o prescan (formada por 8 colunas), no centro temos a parte ativa (que nesta imagem possui 341 colunas) e à direita o overscan (formada por 70 colunas). . . . .	45
Figura 3.8 – Imagem bruta (à esquerda), linhas de base a serem subtraídas (centro) e imagem processada, já com a subtração (à direita). Figura extraída de [21]. . . . .	45
Figura 3.9 – Evolução da constante de calibração ao longo das imagens de um mesmo <i>run</i> para os dois Skipper-CCDs (ACDS-10 e ACDS-11). . . . .	46

Figura 3.10–Exemplo de ajustes gaussianos independentes aos primeiros 20/100 picos da distribuição de dados total (a). Ajuste linear sobre a relação das médias dos 100 primeiros picos em função do número de elétrons (b). . . . .	47
Figura 3.11–Evolução das métricas DNL (a) e INL (b) em função do número de elétrons, para o regime de baixas energias. . . . .	47
Figura 3.12–Eficiência de detecção dos Skipper-CCDs do CONNIE (vermelho), considerando a aceitação na extração de eventos e os cortes de seleção. As eficiências obtidas para os conjuntos de dados de CONNIE com CCDs de 2016-2018 (azul) e de 2019 (cinza) são apresentadas como referência. À esquerda: Eficiência para energias de até 0.5 keV e à direita: um zoom na região de baixas energias. Figura extraída de [37]. . . . .	49
Figura 3.13–Espectros obtidos nos períodos com reator ligado (em vermelho) e desligado (em cinza) e no bloco abaixo a diferença entre eles. Figura extraída de [37]. . . . .	50
Figura 3.14–Comparação entre os limites de exclusão para um mediador vetorial leve com 95% de nível de confiança (C.L.) obtidos pelo CONNIE utilizando Skipper-CCDs e CCDs padrão, e os obtidos pelos experimentos CONUS e COHERENT [46]. Figura extraída de [37]. . . . .	51
Figura 3.15–Limites superiores com 90% de C.L. para interações matéria escura-elétron mediadas por um fóton escuro pesado (à esquerda) e um fóton escuro ultraleve (à direita), estabelecidos por CONNIE (azul, linha sólida), protoSENSEI [53] (laranja, linha sólida), SENSEI [54] (verde, linha pontilhada) e DAMIC-M [55, 56] (vermelho, linha tracejada para buscas com modulação e vermelho, linha pontilhada para buscas diretas). Linhas sólidas correspondem a experimentos realizados na superfície, linhas pontilhadas a limites diretos obtidos no subsolo, e linhas tracejadas a limites modulados obtidos no subsolo. Figura extraída de [37]. . . . .	53
Figura 3.16–Limites de exclusão a 90% de nível de confiança (C.L.) em função da massa e da fração de carga das partículas milicarregadas (mCP) para o CONNIE (verde), Atucha-II (preto) e sua combinação (azul). O limite de exclusão do TEXONO [51] também é incluído para comparação (vermelho). À esquerda: os limites superiores são obtidos a partir da atenuação das mCPs à medida que elas viajam do reator até o detector. À direita: linhas contínuas representam resultados considerando a produção de mCPs tanto por raios $\gamma$ primários quanto secundários, enquanto linhas tracejadas consideram apenas os raios $\gamma$ primários. Figura extraída de [49]. . . . .	53

Figura 4.1 – Imagem de 70 x 140 pixels simulada com ruído de leitura de $0.2 e^-$ (esquerda). Distribuição dos dados desta mesma imagem ajustada por uma função gaussiana (direita). . . . .	57
Figura 4.2 – Imagens simuladas apenas com eventos de elétron único com taxas de $1 e^-/\text{px}/\text{img}$ (à direita), $0.1 e^-/\text{px}/\text{img}$ (no centro) e $0.01 e^-/\text{px}/\text{img}$ (à esquerda). . . . .	58
Figura 4.3 – Passos de simulação do ruído de fundo para uma imagem de 20 x 20 pixels: primeiro simulamos o ruído de leitura (à esquerda), depois a taxa de emissão de elétron único (centro) e, por fim, as duas contribuições são somadas na imagem final (à direita). . . . .	59
Figura 4.4 – Imagem simulada de tamanho 70 x 140 pixels com ruído de leitura igual $0.2 e^-$ e taxa de emissão de elétron único de $0.1 e^-/\text{px}/\text{img}$ (esquerda). Ajuste da convolução sobre a distribuição de dados (direita). . . . .	59
Figura 4.5 – Exemplo das iterações para sorteio de elétron em um evento de 38 eV, que produz 10 elétrons. Os elétrons são somados um a um ao longo das iterações. . . . .	61
Figura 4.6 – Imagem simulada com $\sigma = 0.2 e^-$ e $\lambda = 0.1 e^-/\text{px}/\text{img}$ , com 10 eventos gerados em profundidade de $675 \mu\text{m}$ e energia sorteada no intervalo de $[500,1000]$ eV. . . . .	61
Figura 4.7 – Distribuições das projeções em x e y dos baricentros dos eventos extraídos e catalogados. Os histogramas, acima e à direita, representam as projeções dos baricentros dos eventos nos eixos x e y, respectivamente. . . . .	62
Figura 4.8 – Distribuição das energias reconstruídas para os eventos simulados. O histograma possui excesso no primeiro <i>bin</i> , o que condiz com a extração de eventos falso-positivos, representada pelo gráfico do canto superior direito. . . . .	63
Figura 4.9 – <i>Profile</i> da discrepância relativa entre as energias simulada e reconstruída. . . . .	64
Figura 4.10 – Sobreposição dos espectros real e simulado. . . . .	64
Figura 5.1 – Diagrama ilustrativo dos subconjuntos da inteligência artificial. . . . .	68
Figura 5.2 – Diagrama ilustrativo do processo de aprendizagem de um <i>perceptron</i> para um único vetor de entrada. . . . .	70
Figura 5.3 – Arquitetura de uma rede neural com três camadas ocultas totalmente conectadas (Figura adaptada de <a href="https://doc.comsol.com">https://doc.comsol.com</a> ). . . . .	71
Figura 5.4 – Esquema ilustrativo da aplicação de um filtro de dimensão 2x2 sobre uma matriz (Figura extraída de [63]). . . . .	73
Figura 5.5 – Ilustração dos efeitos do <i>padding</i> na convolução para um <i>kernel</i> 3 x 3. . . . .	74
Figura 5.6 – Ilustração dos efeitos do <i>stride</i> na convolução. . . . .	74
Figura 5.7 – Funções de ativação: Linear, ReLU e <i>Sigmoid</i> . . . . .	76
Figura 5.8 – Ilustração dos efeitos do <i>max pooling</i> sobre os mapas de características. . . . .	77

Figura 5.9 – Ilustração dos efeitos do <i>average pooling</i> sobre os mapas de características.	77
Figura 5.10–Arquitetura típica de uma rede neural convolucional que combina todos os tipos de camadas anteriormente mencionados: camadas de convolução, ativação e agrupamento. Figura adaptada de ( <a href="https://it.mathworks.com/discovery/convolutional-neural-network.html">https://it.mathworks.com/discovery/convolutional-neural-network.html</a> ).	78
Figura 5.11–Representação de como é dividido o <i>dataset</i> : o conjunto de treinamento deve ser o maior dentre os três, seguido do conjunto de teste.	83
Figura 5.12–Esquematização da técnica de validação cruzada: a cada iteração um novo <i>fold</i> é utilizado como conjunto de teste.	84
Figura 6.1 – Gráficos das distribuições de valores mínimo e máximos dos pixels nos eixos X (esquerda) e Y (direita). As linhas tracejadas representam a delimitação de uma imagem de 12 x 12 pixels.	88
Figura 6.2 – Arquitetura do Modelo A.1	89
Figura 6.3 – Curvas de desempenho ao longo do treinamento e validação para o Modelo A.1.	90
Figura 6.4 – Arquitetura do Modelo A.2.	90
Figura 6.5 – Curvas de desempenho ao longo do treinamento e validação para o Modelo A.2.	91
Figura 6.6 – Arquitetura do Modelo A.3.	91
Figura 6.7 – Curvas de desempenho ao longo do treinamento e validação para o Modelo A.3.	91
Figura 6.8 – Arquitetura do Modelo A.4.	92
Figura 6.9 – Curvas de desempenho ao longo do treinamento e validação para o Modelo A.4.	92
Figura 6.10– <i>Profile Plot</i> (a) e <i>Residual Plot</i> (b) para os Modelos A.	93
Figura 6.11–Desvio padrão por <i>bin</i> dos Modelos A.	94
Figura 6.12–Arquitetura base dos Modelos B. Os blocos destacados em amarelo representam as estruturas variadas em cada modelo.	94
Figura 6.13–Curvas de desempenho ao longo do treinamento e validação para o Modelo B.1.	95
Figura 6.14–Curvas de desempenho ao longo do treinamento e validação para o Modelo B.2.	95
Figura 6.15–Curvas de desempenho ao longo do treinamento e validação para o Modelo B.3.	96
Figura 6.16–Curvas de desempenho ao longo do treinamento e validação para o Modelo B.4.	97
Figura 6.17– <i>Profile Plot</i> (a) e <i>Residual Plot</i> (b) para os Modelos B.	98
Figura 6.18–Desvio padrão por <i>bin</i> para os Modelos B.	98

Figura 6.19–Arquitetura base dos Modelos <i>C</i> . O bloco em amarelo destaca o hiper- parâmetro variado nos modelos. . . . .	99
Figura 6.20–Curvas de desempenho ao longo do treinamento e validação para o Modelo <i>C.1</i> . . . . .	99
Figura 6.21–Curvas de desempenho ao longo do treinamento e validação para o Modelo <i>C.2</i> . . . . .	100
Figura 6.22–Curvas de desempenho ao longo do treinamento e validação para o Modelo <i>C.3</i> . . . . .	100
Figura 6.23–Curvas de desempenho ao longo do treinamento e validação para o Modelo <i>C.4</i> . . . . .	101
Figura 6.24– <i>Profile Plot</i> (a) e <i>Residual Plot</i> (b) para os Modelos <i>C</i> . . . . .	102
Figura 6.25–Desvio padrão por <i>bin</i> para os Modelos <i>C</i> . . . . .	102
Figura 6.26–Curvas das funções de <i>loss</i> (a) e métrica (b) por <i>fold</i> para o Modelo <i>C.1</i> . . . . .	103
Figura 6.27– <i>Profile Plot</i> (a) e <i>Residual Plot</i> (b) por <i>fold</i> para o Modelo <i>C.1</i> . . . . .	104
Figura 6.28–Curvas das funções de <i>loss</i> (a) e métrica (b) por <i>fold</i> para o Modelo <i>C.2</i> . . . . .	105
Figura 6.29– <i>Profile Plot</i> (a) e <i>Residual Plot</i> (b) por <i>fold</i> para o Modelo <i>C.2</i> . . . . .	105
Figura 6.30–Curvas das funções de <i>loss</i> (a) e métrica (b) por <i>fold</i> para o Modelo <i>C.3</i> . . . . .	106
Figura 6.31– <i>Profile Plot</i> (a) e <i>Residual Plot</i> (b) por <i>fold</i> para o Modelo <i>C.3</i> . . . . .	106
Figura 6.32–Curvas das funções de <i>loss</i> (a) e métrica (b) por <i>fold</i> para o Modelo <i>C.4</i> . . . . .	107
Figura 6.33– <i>Profile Plot</i> (a) e <i>Residual Plot</i> (b) por <i>fold</i> para o Modelo <i>C.4</i> . . . . .	108
Figura 6.34–Arquitetura base dos Modelos <i>C</i> . Os blocos em amarelo representam as estruturas variadas nos modelos. . . . .	109
Figura 6.35–Curvas de desempenho ao longo do treinamento e validação para o Modelo <i>D.1</i> . . . . .	109
Figura 6.36–Curvas de desempenho ao longo do treinamento e validação para o Modelo <i>D.2</i> . . . . .	110
Figura 6.37–Curvas de desempenho ao longo do treinamento e validação para o Modelo <i>D.3</i> . . . . .	110
Figura 6.38–Curvas de desempenho ao longo do treinamento e validação para o Modelo <i>D.4</i> . . . . .	111
Figura 6.39– <i>Profile Plot</i> (a) e <i>Residual Plot</i> (b) para os melhores modelos das rodadas <i>C</i> e <i>D</i> . . . . .	112
Figura 6.40–Desvio padrão por <i>bin</i> para os melhores modelos das rodadas <i>C</i> e <i>D</i> . . . . .	112
Figura 6.41–Arquitetura base dos Modelos <i>E</i> . Estão destacados em amarelo os blocos de <i>dropout</i> , cujas taxas são variadas em cada modelo. . . . .	113
Figura 6.42–Curvas de desempenho ao longo do treinamento e validação para o Modelo <i>E.1</i> . . . . .	113
Figura 6.43–Curvas de desempenho ao longo do treinamento e validação para o Modelo <i>E.2</i> . . . . .	114



Figura 6.44–Curvas de desempenho ao longo do treinamento e validação para o Modelo E.3. . . . .	114
Figura 6.45–Curvas de desempenho ao longo do treinamento e validação para o Modelo E.4. . . . .	115
Figura 6.46–Curvas de desempenho ao longo do treinamento e validação para o Modelo E.5. . . . .	116
Figura 6.47–Curvas de desempenho ao longo do treinamento e validação para o Modelo E.6. . . . .	116
Figura 6.48– <i>Profile Plot</i> (a) e <i>Residual Plot</i> (b) para os modelos E. . . . .	117
Figura 6.49–Desvio padrão por <i>bin</i> para os modelos E. . . . .	117
Figura 6.50–Curvas de desempenho ao longo do treinamento e validação para o Modelo F.1. . . . .	118
Figura 6.51–Curvas de desempenho ao longo do treinamento e validação para o Modelo F.2. . . . .	119
Figura 6.52–Curvas de desempenho ao longo do treinamento e validação para o Modelo F.3. . . . .	119
Figura 6.53–Curvas de desempenho ao longo do treinamento e validação para o Modelo F.4. . . . .	120
Figura 6.54– <i>Profile Plot</i> (a) e <i>Residual Plot</i> (b) para os modelos F. . . . .	121
Figura 6.55–Desvio padrão por <i>bin</i> para os Modelos F. . . . .	121
Figura 6.56–Curvas de desempenho ao longo do treinamento e validação para o Modelo G.1. . . . .	122
Figura 6.57–Curvas de desempenho ao longo do treinamento e validação para o Modelo G.2. . . . .	122
Figura 6.58–Curvas de desempenho ao longo do treinamento e validação para o Modelo G.3. . . . .	123
Figura 6.59– <i>Profile Plot</i> (a) e <i>Residual Plot</i> (b) para os modelos G. . . . .	124
Figura 6.60–Desvio padrão por <i>bin</i> para os Modelo G. . . . .	124
Figura 6.61– <i>Profile Plot</i> (a) e <i>Residual Plot</i> (b) comparativos entre os resultados de predição do algoritmo de CONNIE e a rede neural. . . . .	125
Figura 6.62–Desvio padrão por <i>bin</i> para o algoritmo usual de CONNIE e a rede neural G.2. . . . .	125
Figura 7.1 – Diagrama da arquitetura G.2, que apresentou o melhor desempenho em nossos estudos (seção 6.3). . . . .	129
Figura A.1–Reconstrução de eventos falso positivos para diferentes valores de ruído de leitura e taxa de emissão de elétron único. . . . .	132



Figura A.2 – Reconstrução de eventos falso positivos focada nos valores de $\sigma$ iguais a 0.10, 0.16 e 0.2 $e^-$ .	132
---	-----

## Lista de tabelas

Tabela 1 – Hiperparâmetros utilizados no treinamento das redes neurais até a rodada E.	89
Tabela 2 – Valores da função de perda e métrica da validação para os modelos da rodada A.	93
Tabela 3 – Valores da função de perda e métrica da validação para os modelos da rodada B.	97
Tabela 4 – Valores da função de perda e métrica da validação para os modelos da rodada C.	101
Tabela 5 – Valores da função de perda e métrica para diferentes <i>folds</i> para o modelo C.1	104
Tabela 6 – Valores da função de perda e métrica para diferentes <i>folds</i> para o modelo C.2	105
Tabela 7 – Valores da função de perda e métrica para diferentes <i>folds</i> para o modelo C.3	107
Tabela 8 – Valores da função de perda e métrica para diferentes <i>folds</i> para o modelo C.4	108
Tabela 9 – Valores da função de perda e métrica da validação para os modelos da rodada D.	111
Tabela 10 – Valores da função de perda e métrica da validação para os modelos da rodada E.	117
Tabela 11 – Valores da função de perda e métrica da validação para os modelos da rodada F.	120
Tabela 12 – Valores da função de perda e métrica da validação para os modelos da rodada G.	123

# Capítulo 1

## Introdução

A existência dos neutrinos foi postulada em 1930, por Wolfgang Pauli, a partir de seus estudos do decaimento beta. Nesta desintegração, um nêutron dentro de um núcleo instável se transmuta em um próton e emite um elétron. As medições da época mostraram que o espectro de energia do elétron era contínuo, diferentemente do que se esperava para um processo de dois corpos. Isso sugeria que a energia não estava sendo conservada, o que intrigou à comunidade científica. Para resolver o problema, Pauli [1] postulou a existência de uma partícula neutra, sem massa e com spin  $1/2$ , que passava despercebida pelos detectores. Anos mais tarde, Enrico Fermi [2] formalizou a teoria e batizou essas partículas misteriosas de neutrinos.

Os neutrinos são partículas elementares extremamente abundantes no Universo, sendo superados em número apenas pelos fótons. A cada segundo, trilhões deles, que vem principalmente do Sol [3], atravessam o corpo humano. Mas, devido às suas propriedades, interagem raramente com a matéria, o que torna sua detecção um grande desafio na física experimental. O Modelo Padrão identifica as partículas elementares que constituem a matéria e descreve como elas interagem [4]. De acordo com essa teoria, os neutrinos são léptons e podem ser classificados em três sabores: neutrino do elétron,  $\nu_e$ , neutrino do múon,  $\nu_\mu$ , e neutrino do tau,  $\nu_\tau$ , além das respectivas antipartículas  $\bar{\nu}_e$ ,  $\bar{\nu}_\mu$  e  $\bar{\nu}_\tau$ .

Interagindo apenas através da força fraca, acreditava-se que os neutrinos atravessariam invisíveis qualquer aparato de detecção. Essa visão mudou em 1956, quando Frederick Reines e Clyde Cowan observaram, pela primeira vez, antineutrinos do elétron via decaimento beta inverso [5]. O neutrino do múon foi detectado anos mais tarde, em 1962, num experimento com feixes de píons [6], e a última descoberta, em 2000, foi o neutrino do tau pela colaboração DONUT [7]. Estes marcos consolidaram o Modelo Padrão como uma teoria bem-sucedida, mas, ao mesmo tempo, o estudo detalhado das propriedades dos neutrinos levantou novas questões que o modelo atual não consegue responder.

No final dos anos 60, o experimento *Homestake* [8], observou um fluxo de neutrinos solares muito menor do que o previsto pelo Modelo Solar Padrão. Esta discrepância ficou conhecida como *Problema do Neutrino Solar*, e permaneceu sem explicação por décadas.

O físico Bruno Pontecorvo [9] idealizou uma teoria ambiciosa onde os neutrinos poderiam oscilar, mudando de sabor ao percorrer longas distâncias, o que explicaria a diferença de fluxo de neutrinos medida. Para que isto fosse possível, estas partículas deveriam possuir massa não nula. Nos anos 2000, uma análise colaborativa dos resultados dos experimentos Super Kamiokande [10], Sudbury Neutrinos Observatory (SNO) [11] e KAMland [12] comprovou a hipótese de Pontecorvo, mostrando que os neutrinos possuem massa, mesmo que muito pequena.

O Modelo Padrão não explica o mecanismo de massa dos neutrinos, e esta é uma das evidências que esta teoria precisa ser ampliada para contemplar questões ainda em aberto. Neste contexto, as interações de neutrinos se apresentam como ferramentas úteis para validar o atual Modelo e compreender a física que se estende para além dele. Estas interações ocorrem através de canais que podem ser categorizados como corrente carregada, quando há troca de carga entre o neutrino e a partícula alvo, e corrente neutra, onde não há troca de carga e o neutrino interatante não muda de sabor.

Postulado em 1973 [13], o espalhamento elástico coerente neutrino-núcleo ocorre quando o neutrino interage com o núcleo como um todo, sem mudar a organização dos núcleons [14]. A transferência de momento envolvida na interação é pequena a ponto de tornar o comprimento de onda associado maior que o núcleo. Neste regime, as funções de onda dos núcleons estão em fase, o que contribui para o aumento da seção de choque da interação, que cresce com o quadrado do número de nêutrons [15, 16]. A seção de choque diferencial da interação para um núcleo de massa  $M$  pode ser descrita pela equação 1.1, em termos da energia do neutrino incidente  $E_\nu$ . Nela,  $G_F$  é a constante de Fermi e  $F_W(q^2)$  representa o fator de forma fraco. A carga fraca  $Q_W$ , expressa pela equação 1.2, é obtida em função do número de prótons ( $Z$ ) e nêutrons ( $N$ ) dos núcleos, e de  $\sin \theta_W$ , que representa o ângulo de mistura fraco.

$$\frac{d\sigma}{dE_R} = \frac{G_F^2 M}{4\pi} \left(1 - \frac{ME_R}{2E_\nu^2}\right) Q_W^2 F_W^2(q^2), \quad (1.1)$$

$$Q_W = Z(1 - 4\sin^2 \theta_W) - N. \quad (1.2)$$

Este regime de coerência só é preservado para neutrinos com energias menores que aproximadamente 50 MeV, acima deste valor o comprimento de onda torna-se menor que o tamanho do núcleo, e esta propriedade é perdida.

A energia máxima do recuo nuclear pode ser expressa em termos da energia do neutrino incidente e da massa do núcleo, como:

$$E_R^{\max} = \frac{2E_\nu^2}{(M + 2E_\nu)}. \quad (1.3)$$

O maior desafio para detectar  $\text{CE}\nu\text{NS}$  vem do fato de que o recuo nuclear resultante produz sinais com energias menores que algumas poucas dezenas de keV. No entanto, apenas uma parte desta pequena energia de recuo contribui efetivamente para ionização do sensor, e, conseqüentemente, para a produção de portadores de carga detectáveis. A quantidade de energia de recuo que resulta na ionização é medida pelo fator de *quenching* [17]. Além disso, parte da energia pode ser dissipada por outros canais, como excitação atômica de gerações de fónons, que também podem ser explorados em técnicas alternativas de detecção. Por este motivo, a medição de  $\text{CE}\nu\text{NS}$  requer detectores com alta sensibilidade à eventos de baixas energias.

O espalhamento elástico coerente neutrino-núcleo foi detectado, pela primeira vez em 2017, pelo experimento COHERENT [18], que utilizou detectores cintiladores CsI[Na]. A fonte de neutrinos desta medição foi o acelerador Spallation Neutron Source (SNS). Nele, prótons de altas energias são acelerados e, ao colidirem com um alvo, produzem píons. Estes mésons desaceleram e decaem dando origem a múons e seus neutrinos correspondentes (conforme reação 1.4). Os múons, por sua vez, decaem em pósitrons, neutrinos do elétron e antineutrinos do múon (de acordo com a reação 1.5). A energia dos neutrinos do múon é concebida num pico monoenergético de aproximadamente 30 MeV, enquanto os demais sabores possuem um espectro contínuo de energia, com o máximo em 58.2 MeV [19].

$$\pi^+ \longrightarrow \nu_\mu + \mu^+ \quad (1.4) \quad \mu^+ \longrightarrow e^+ + \nu_e + \bar{\nu}_\mu \quad (1.5)$$

Um outro tipo de instalação que é explorada na busca por  $\text{CE}\nu\text{NS}$  são os reatores nucleares. Um reator pode gerar energia através do processo de fissão nuclear de quatro isótopos:  $^{235}\text{U}$ ,  $^{238}\text{U}$ ,  $^{239}\text{Pu}$ ,  $^{241}\text{Pu}$ . Durante este processo, fragmentos radioativos com excesso de nêutrons são produzidos e passam por consecutivos decaimentos beta, que são expressos pela reação 1.6.

$$n \longrightarrow p + e^- + \bar{\nu}_e \quad (1.6)$$

Além disto, a captura de nêutrons pelo  $^{238}\text{U}$  também contribui para o fluxo de neutrinos. Neste processo, o núcleo absorve um nêutron e fica excitado emitindo um fóton  $\gamma$ . Depois, o núcleo produzido pela captura,  $^{239}\text{U}$ , sofre decaimento beta, emitindo um elétron e seu antineutrino correspondente. O fluxo de antineutrinos de reator está reportado na figura 1.1. A energia média dos antineutrinos produzidos é de 1 MeV, e o pico observado entorno de 0.5 MeV é proveniente de reações de captura de nêutrons [20].

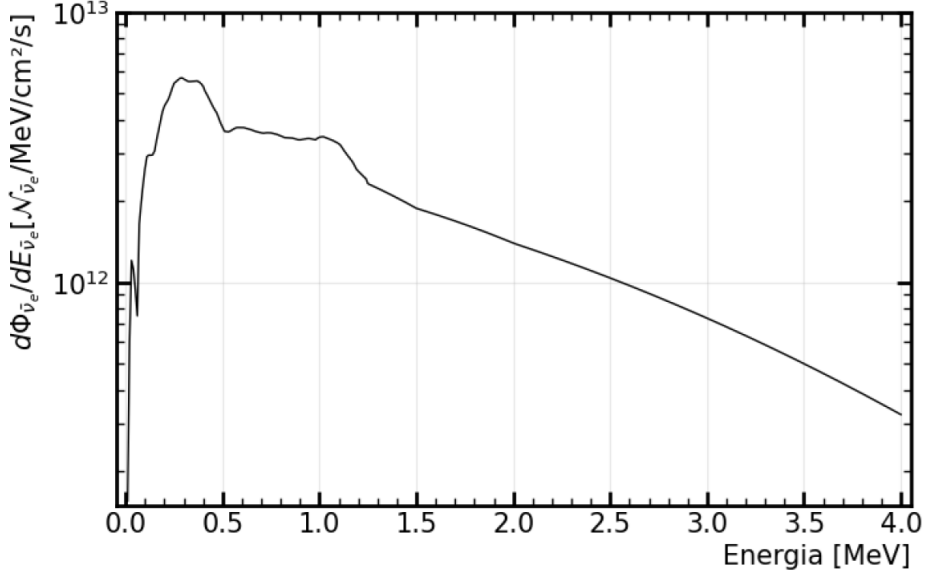


Figura 1.1 – Fluxo de antineutrinos do elétron em função da energia calculado para a posição onde o experimento CONNIE está instalado, há 30 metros do reator nuclear da usina de Angra 2. Figura extraída de [21].

Devido às baixas energias dos antineutrinos emitidos por reatores, o limiar de detecção implementado no aparato experimental precisa ser ainda menor do que o utilizado para neutrinos de aceleradores. Uma técnica promissora para esse fim é a utilização de sensores CCDs (Charge Coupled Devices). Estes sensores de silício, formados por uma matriz de capacitores acoplados vêm ganhando espaço na física de partículas, por seu baixíssimo ruído de leitura. O experimento CONNIE (Coherent Neutrino Nucleus Interaction Experiment) foi o primeiro a empregar CCDs na busca pela detecção de  $CE\nu NS$  com antineutrinos de reator [22]. Ele opera desde 2014 na Central Nuclear Almirante Álvaro Alberto, em Angra dos Reis, e está instalado a 30 metros do reator nuclear de Angra 2.

Esta dissertação está organizada conforme a seguir. No capítulo 2 vamos discutir os princípios de funcionamento de um CCD e suas principais características para operar como detector de partículas. Também vamos apresentar os avanços significativos que o desenvolvimento de novos sensores, os Skipper-CCDs, trouxeram a este cenário. No capítulo 3 vamos relatar o aparato experimental do CONNIE, comentando as diferentes fases de operação com CCDs e Skipper-CCDs. Será apresentada a cadeia de processamento e análise dos dados das imagens, incluindo a metodologia de calibração que desenvolvi junto com o estudo de sua linearidade. Neste capítulo também serão reportados os resultados mais recentes do experimento, obtidos com Skipper-CCDs. Para selecionar eventos de modo a reduzir o ruído de fundo presente no experimento, propomos a utilização de uma rede neural convolucional para estimar a profundidade dos eventos de carga pontual. Desenvolvemos um algoritmo para simular as imagens utilizadas para treinar e validar a rede, o que será detalhado no capítulo 4. Uma breve revisão sobre o funcionamento das redes neurais convolucionais é apresentada no capítulo 5. A busca pela arquitetura mais adequada para

o problema é detalhada no capítulo 6, bem como a comparação entre as performances da rede escolhida e o algoritmo utilizado atualmente em CONNIE. No capítulo 7, faremos as considerações finais acerca do desempenho da rede neural convolucional e vamos delinear as contribuições e perspectivas futuras do presente trabalho.



## Capítulo 2

# Dispositivos de Carga Acoplada

Os CCDs foram projetados em 1969 como dispositivos dedicados ao armazenamento de memória. Os físicos idealizadores, Willard Boyle e George Smith, combinaram o princípio de funcionamento das memórias de bolha magnética à estrutura dos transistores [23]. De forma equivalente a uma tecnologia de armazenamento, as informações guardadas pelos CCDs são pacotes de carga, que são transferidos e lidos um a um pelo amplificador através de um processo de deslocamento sequencial. Logo, a alta performance destes sensores, caracterizada pela boa linearidade e baixo ruído de leitura, motivou sua aplicação na criação de imagens. O constante aprimoramento dos CCDs permitiu que fossem utilizados para fins astronômicos, compondo sistemas de câmera de telescópios [24, 25]. Mais recentemente, estes dispositivos começaram a ser empregados como detectores de partículas, em experimentos que buscam por interações de baixas energias, como é o caso de DAMIC [26], que busca detectar matéria escura de forma direta, e CONNIE [27].

Os menores elementos dos CCDs são pixels formados por capacitores do tipo metal-óxido-semicondutor (MOS). Estes pequenos componentes são acoplados e dispostos em um arranjo bidimensional, na forma de uma matriz. Os capacitores MOS são formados por três camadas: a primeira é o substrato semicondutor, constituídos por uma junção PN em silício; a camada intermediária é um dielétrico, que atua como um filme isolante; e a terceira é um eletrodo metálico, onde as tensões serão aplicadas. Esta última camada, fabricada em silício policristalino, possui estrutura trifásica, com três *gates* conectados a fases de tensão distintas, como ilustrado na figura 2.1 (a).

Para serem utilizados em experimentos que buscam detectar matéria escura e neutrinos, os CCDs precisam ter certas características, tanto no material utilizado para fabricação quanto no modo de operação, com o intuito de otimizar o desempenho. Diferentemente dos dispositivos comuns, estes CCDs são construídos em silício de alta resistividade, da ordem de  $18 \text{ k}\Omega\text{-cm}$ , o que permite sua total depleção. Além disto, CCDs de uso científico operam com retroiluminação (*back-illumination*), assim a luz incide na superfície oposta aos *gates* e interage diretamente com a parte ativa dos pixels. Estas modificações fornecem aos CCDs mais sensibilidade e alta eficiência quântica. Para reduzir o ruído, os sensores científicos



combinam um lento processo de leitura e a diminuição de capacitâncias parasitas. Um outro fator preponderante para a performance é a corrente escura, proveniente da emissão térmica espontânea de portadores de carga. Para minimizar os efeitos desta contribuição os dispositivos operam em baixíssimas temperaturas, da ordem de 100 K.

Neste capítulo vamos detalhar os fundamentos da operação dos CCDs, desde a criação e armazenamento das cargas até o processo de leitura. Apresentaremos os princípios de funcionamento dos Skipper-CCDs, em especial o processo de leitura não destrutivo, que distingue essa nova geração de sensores dos convencionais e permite alcançar níveis de ruído subeletrônicos. Também faremos uma discussão sobre o processo de difusão das cargas, em particular para eventos de carga pontual.

## 2.1 CRIAÇÃO E ARMAZENAMENTO DAS CARGAS

A geração de cargas se dá através da produção de pares elétron-buraco, decorrentes da ionização produzida no sensor. A energia mínima para que um par elétron-buraco seja criado no silício é de 3.745 eV [28], para temperaturas da ordem de 100 K. A interação de um fóton no volume ativo do detector cria cargas via efeito fotoelétrico [29]. Depois de criados, os portadores de carga são arrastados até a zona de depleção, que aparece na interface da junção PN devido a aplicação de uma tensão de polarização nos *gates*, conforme figura 2.1 (b).

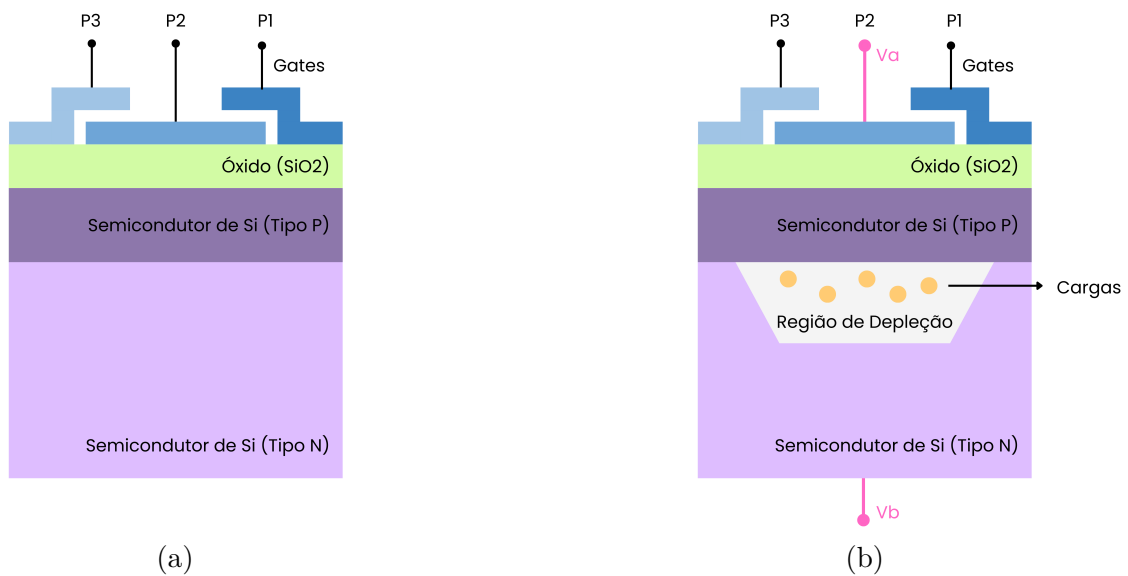


Figura 2.1 – Estrutura de um pixel MOS, formado por uma junção PN, uma camada de óxido e os *gates* (a). Cargas armazenadas na região de depleção em decorrência da aplicação de uma tensão de polarização nos *gates* (b). Representações pictóricas de um pixel MOS, com proporções reais não preservadas.

Partículas carregadas, como elétrons e múons, atravessam o sensor e deixam um rastro de ionização ao longo de suas trajetórias. Quando um antineutrino ou uma partícula de matéria escura interage com os núcleos de silício dos detectores, eles produzem um recuo que, por sua vez, gera ionização com eficiência dependente do fator de *quenching* [17]. Isto induz a criação de pares elétron-buraco, de acordo com o que é ilustrado na figura 2.2. Outras interações também podem promover recuos nucleares, como as decorrentes de nêutrons, por isso, estas partículas atuam como um *background* irreduzível para experimentos como CONNIE e DAMIC.

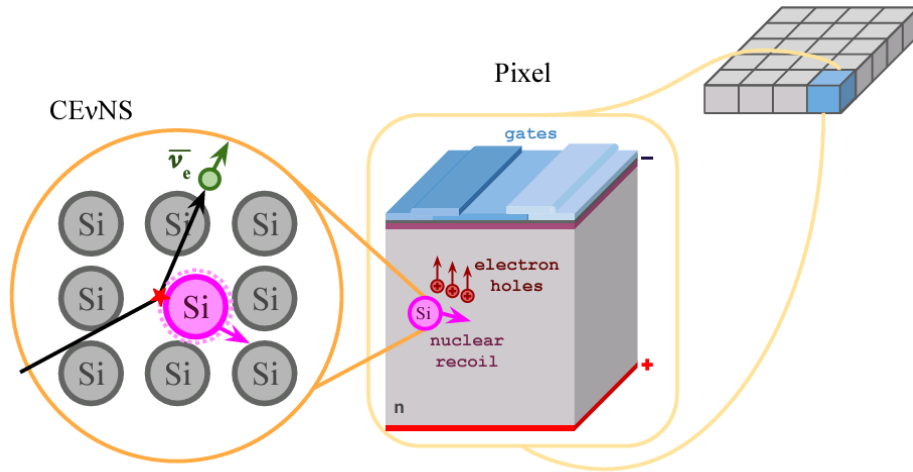


Figura 2.2 – Processo de geração de cargas no volume ativo do detector. Vide detalhes no texto. Fonte: Colaboração CONNIE.

## 2.2 PROCESSO DE LEITURA SEQUENCIAL

O grande sucesso dos CCDs é decorrente de um procedimento inovador de transferência de cargas, que permite uma leitura sequencial delas. Na figura 2.3 está representado este mecanismo de leitura sequencial para um CCD de 3 x 3 pixels. A transferência é controlada pela aplicação ordenada de tensões nos *gates*, que formam uma porta trifásica ( $P_1$ ,  $P_2$  e  $P_3$ ). Quando  $P_2$  está em alta tensão e  $P_1$  e  $P_3$  em baixa, cria-se um poço potencial sob  $P_2$  que aprisiona as cargas (estado 1). Neste caso,  $P_2$  funciona como uma fase coletora enquanto  $P_1$  e  $P_3$  vão operar como fases de barreira, que manterão os portadores de carga aprisionados no poço. Ou seja, as cargas produzidas pelas diferentes interações no volume do sensor são armazenadas nos *gates*. Para a leitura sequencial, a carga armazenada nos pixels de cada fileira horizontal será deslocada para a fileira inferior sendo esta última deslocada para registro horizontal. Este processo é realizado mediante a variação controlada das tensões aplicadas, como representado do lado direito da figura 2.3. Primeiro baixamos o potencial de  $P_1$  para que o pacote de carga fique compartilhado entre o  $P_2$  e  $P_1$  do mesmo pixel (estado 2). Em seguida, eleva-se novamente  $P_2$ , de modo que o pacote de cargas passe

completamente para o poço de potencial de  $P_1$  (estado 3). Depois, reduz-se o potencial de  $P_3$  do pixel seguinte, e assim sucessivamente, até que, finalmente, no estado 7 o pacote de carga esteja sob  $P_2$  do pixel seguinte, como indicado na figura.

As cargas são impedidas de se moverem para a coluna vizinha por conta de uma barreira de potencial (*channel stop*), que é colocada entre cada arranjo vertical de pixels. A transferência é repetida até movimentar todas as cargas de uma mesma linha para o registro horizontal (*serial register*). Neste estágio de leitura, disposto de forma perpendicular as linhas da matriz pixelizada, os pacotes de carga são movimentados, através do mesmo mecanismo de tensões alternadas nas portas *gates*, um a um, mudando as tensões dos potenciais  $H$  ( $H_1, H_2, H_3$ ), até o amplificador. Antes de cada pacote de carga ser transferido para o nó de detecção *sense node*, mede-se a tensão de referência do circuito. Em seguida, o pacote de carga do pixel é movimentado até o nó de detecção, provocando uma variação de potencial proporcional ao número de portadores armazenados. A diferença entre essa leitura e a tensão de referência fornece um sinal diretamente proporcional à carga no pixel. Esse sinal é então amplificado e digitalizado, completando o processo de leitura de cada pixel. Uma vez lido todo o registro horizontal, o procedimento completo se repete para a próxima linha de pixels. Dessa forma, sucessivamente, são transferidas e lidas todas as cargas coletadas na região ativa do sensor.

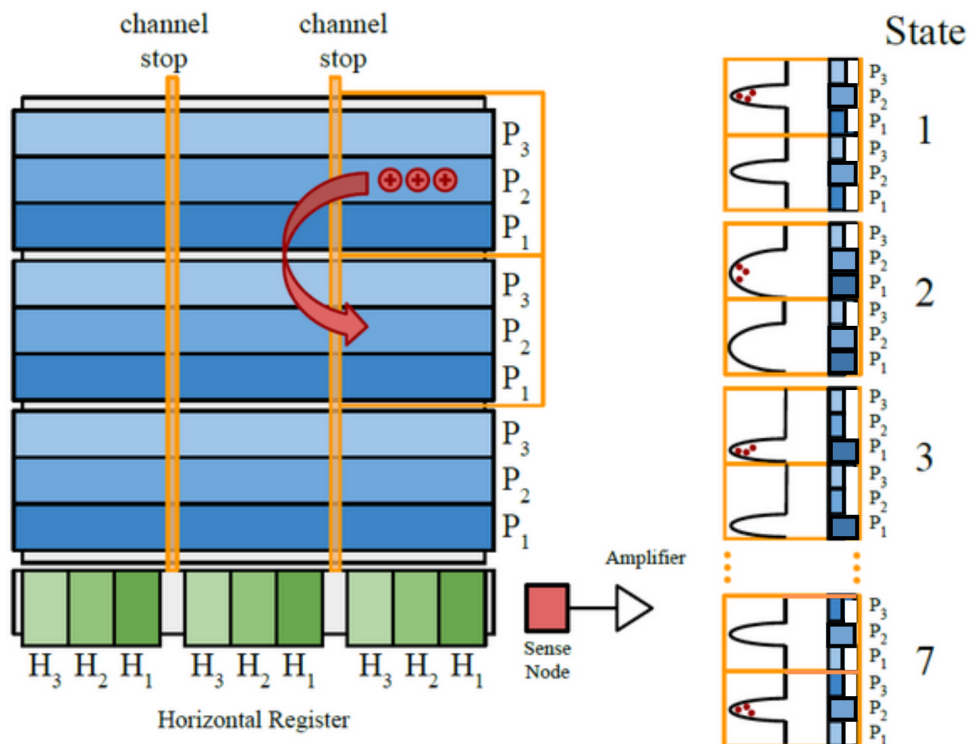


Figura 2.3 – Processo de leitura sequencial das cargas, onde os pacotes de carga dos pixels são movidos linha a linha por meio dos potenciais  $P_i$ , e quando no registro horizontal, são movimentados com os potenciais  $H_i$  até o amplificador. Figura alterada de [21].

Como veremos na seção a seguir, o resultado da exposição e leitura dos sensores é uma imagem em que a distribuição de cargas dos pixels vazios pode ser modelada por uma função gaussiana centrada em zero, cujo desvio padrão representa o ruído de leitura. Tipicamente, este desvio é de 2 a 3 elétrons, o que configura um ruído relativamente baixo para um detector de partículas. No entanto, na busca por interações de muito baixas energias, como CE $\nu$ NS, este ruído de leitura atua como um limiar de detecção, pois não é possível distingui-lo dos eventos de carga pontual. Por este motivo, uma nova geração de sensores, chamados de Skipper-CCDs, que serão discutidos mais a fundo na seção 2.4, foi idealizada com a proposta de reduzir ainda mais o ruído de leitura, inaugurando um novo patamar de sensibilidade no âmbito das tecnologias de detecção.

## 2.3 IMAGENS DE UM CCD E EXTRAÇÃO DE EVENTOS

A exposição e leitura destes sensores resulta em uma imagem do tipo FITS (Flexible Image Transport System) [30] com as assinaturas características de cada interação. Os eventos são identificados pelo traço geométrico e a energia depositada. Como pode ser observado na figura 2.4, os elétrons deixam padrão sinuoso, o que é ocasionado pelos múltiplos espalhamentos que ele sofre. Os múons, por sua vez, são mais massivos, por isso atravessam o sensor ionizando e deixando um traçado linear. Embora pouco frequentes, as partículas alfa são massivas e duplamente carregadas, estas propriedades combinadas resultam na deposição concentrada de energias da ordem de 5 MeV em uma região circular. Um outro tipo de sinal presente nas imagens são os eventos de carga pontual. Eles são decorrentes de interações de baixas energias ocasionadas por fótons, emissões de elétron único associadas à corrente escura e, inclusive, recuos nucleares produzidos por nêutrons e antineutrinos.

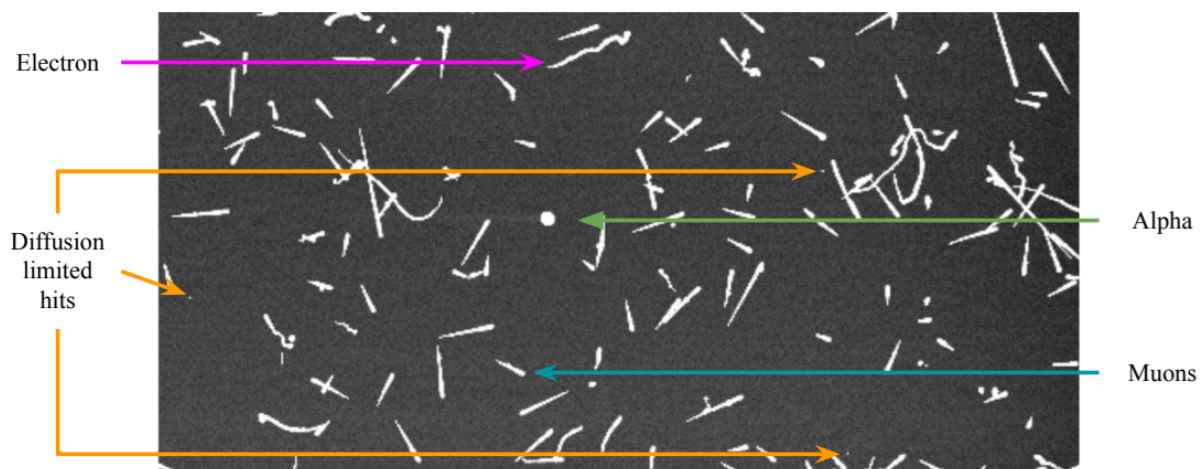


Figura 2.4 – Imagem FITS com diferentes assinaturas características. Figura extraída de [21].

Cada evento pode ser entendido como um aglomerado de pixels associados à deposição de energia nos CCDs. A reconstrução começa com a identificação de pixels adjacentes cuja carga seja maior que um certo limiar; estes pixels são chamados de sementes (nível 0). A identificação de uma semente é feita com base em um limiar, normalmente fixado em  $4\sigma$  (onde  $\sigma$  é o ruído de leitura associado à imagem) para evitar que eventos de ruído de fundo sejam criados devido a flutuações positivas. A partir das sementes, são adicionadas camadas de pixels vizinhos sem exigência de limiar adicional: os pixels de nível 1 correspondem a todos os vizinhos (inclusive na diagonal) em contato com os pixels semente; os pixels de nível 2 são aqueles em contato com os de nível 1, e assim sucessivamente. Para níveis superiores a 2, apenas os vizinhos adjacentes são considerados. O aglomerado é definido como a união de todos os pixels pertencentes a essas camadas.

Os eventos extraídos são guardados num catálogo com formato de arquivo root [31]. Na figura 2.5 podemos observar alguns exemplos representativos destes eventos extraídos: da esquerda para a direita, um múon, um evento de carga pontual e um elétron. Para cada evento, além de guardar o vetor de coordenadas com a carga em cada pixel, são armazenados também os observáveis que caracterizam eles, como por exemplo, a energia total calculada como a soma de todas as energias associadas aos pixels envolvidos, a posição do baricentro, que é determinada através das equações 2.1 e 2.2, tendo em conta as posições do  $i$ -ésimo pixel do evento  $(x_i, y_i)$ , e a energia associada a esse pixel  $E_i$ . Também calculamos um observável baseado na largura projetada de cada evento nos eixos,  $(xVar, yVar)$ , que utilizamos, por exemplo, para caracterizar a natureza dos eventos.

$$x_{bary} = \frac{\sum_i x_i E_i}{\sum_i E_i} \quad (2.1) \quad y_{bary} = \frac{\sum_i y_i E_i}{\sum_i E_i} \quad (2.2)$$

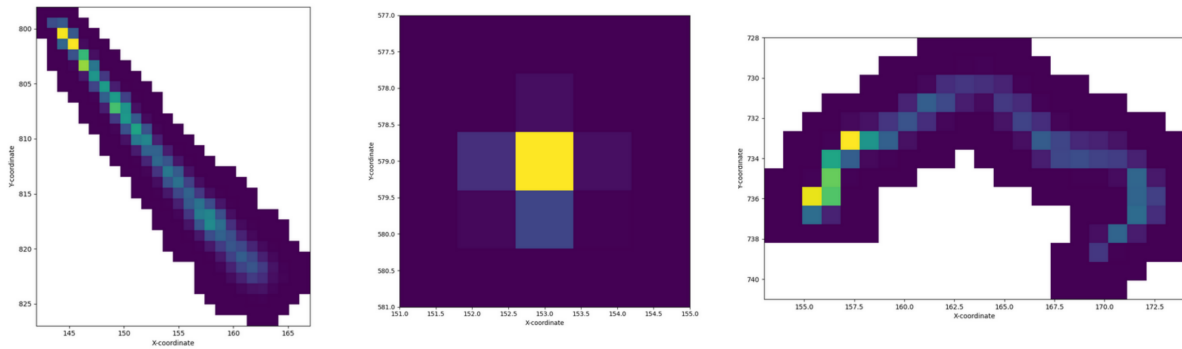


Figura 2.5 – Eventos extraídos e catalogados. À esquerda temos um traço de um múon, ao centro um evento de carga pontual e à direita a assinatura geométrica de um elétron.

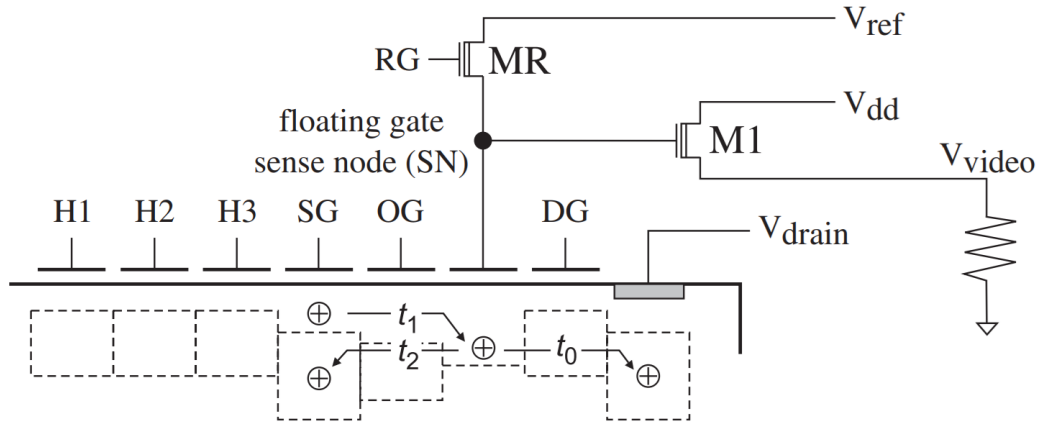


Figura 2.6 – Esquematisação do processo de leitura não destrutivo de um Skipper-CCD. Figura extraída de [32].

## 2.4 SKIPPER-CCDs

Os Skipper-CCDs utilizam os mesmos princípios de funcionamento que os CCDs tradicionais, com o mesmo mecanismo de deslocamento sequencial das cargas. O diferencial destes novos sensores é que eles implementam um processo de leitura não destrutivo, o que permite que as cargas de cada pixel sejam lidas repetidas vezes. Isto é viabilizado por um estágio de porta flutuante (*Summing Well*, SG) que armazena temporariamente os pacotes de carga antes que sejam transferidos para o nó de detecção (*sense node*, SN), onde são lidos. Este sistema de leitura é dividido em três etapas, indicadas no esquema da figura 2.6 por  $t_0$ ,  $t_1$  e  $t_2$ . No instante  $t_0$  o pacote de cargas no nó de detecção é movido para o  $V_{\text{drain}}$  para ser descartado, isto é feito através de um estágio de porta de descarga (*drain gate*, DG). Enquanto isso, a tensão do nó de detecção é restaurada para o valor de referência ( $V_{\text{ref}}$ ) através de um estágio de *reset* (RG), garantindo que a leitura do próximo pacote seja feita sem influência de cargas residuais. No instante  $t_1$  o pacote de cargas acumulado na porta flutuante, proveniente dos estágios do registro horizontal  $H_1$ ,  $H_2$  e  $H_3$ , é transferido através da porta de saída (*output gate*, OG) para o nó de detecção, onde é lido. Esta leitura configura a primeira amostra de carga do pixel. No instante  $t_2$  o pacote de cargas retorna para a porta flutuante para que a tensão do nó de detecção seja reinicializada para o valor de referência. Este procedimento é repetido consecutivas vezes para tomar  $N$  amostras da carga de um mesmo pixel. Uma vez alcançado o valor  $N$ , um novo ciclo começa para ler o próximo pixel do sensor.

O princípio estatístico que fundamenta os Skipper-CCDs afirma que, ao repetir a leitura da mesma carga diversas vezes, o valor final do pixel pode ser obtido pela média dessas amostras. Desta forma, o ruído associado ao processo de leitura diminui com o número de repetições, seguindo uma lei que decresce com a raiz quadrada de  $N$ , o número de amostras. Essa relação é expressa pela equação 2.3, onde  $\sigma_1$  corresponde ao ruído de uma única leitura e  $\sigma_N$  ao ruído efetivo após  $N$  amostras do mesmo pixel.



$$\sigma_N = \frac{\sigma_1}{\sqrt{N}} \quad (2.3)$$

De acordo com esta regra, quanto maior o número de amostras menor será o ruído. A leitura não destrutiva de cargas permite que os Skipper-CCDs alcancem ruídos ultrabaixos, da ordem de subelétrons a depender do número de amostras, o que resulta em imagens com excelentes resoluções, como pode ser observado na figura 2.7. Com 4 amostras, reconstruímos a melhor performance já obtida com um CCD padrão, com ruído de aproximadamente  $2\text{ e}^-$ . Mas aumentando significativamente a quantidade de vezes que lemos cada pixel, 400 no caso da última imagem da figura 2.7, alcançamos um ruído de leitura de  $0.15\text{ e}^-$ .

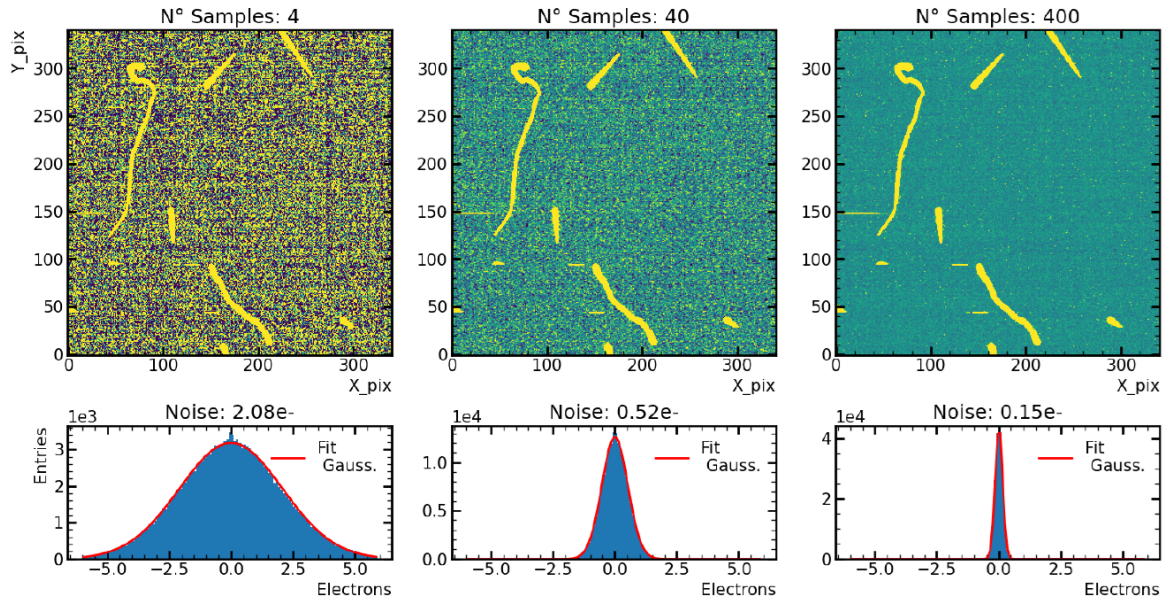


Figura 2.7 – Evolução da resolução de imagens de Skipper-CCDs produzidas com 4 (à esquerda), 40 (ao centro) e 400 (à direita) amostras por pixel. Figura extraída de [21].

Um dos desdobramentos do processo de leitura implementado nos Skipper-CCDs pode ser contemplado no histograma de distribuição de carga nos pixels. Por conta do ruído de leitura subeletrônico, estes sensores adquirem a capacidade de 'contar' os elétrons provenientes das interações. Por isto, a distribuição tem perfil de picos centrados no respectivo número de elétrons, como mostra a figura 2.8. A largura dos picos observados corresponde ao ruído de leitura obtido para o sensor, que é igual a  $0.16\text{ e}^-$  no nosso caso.

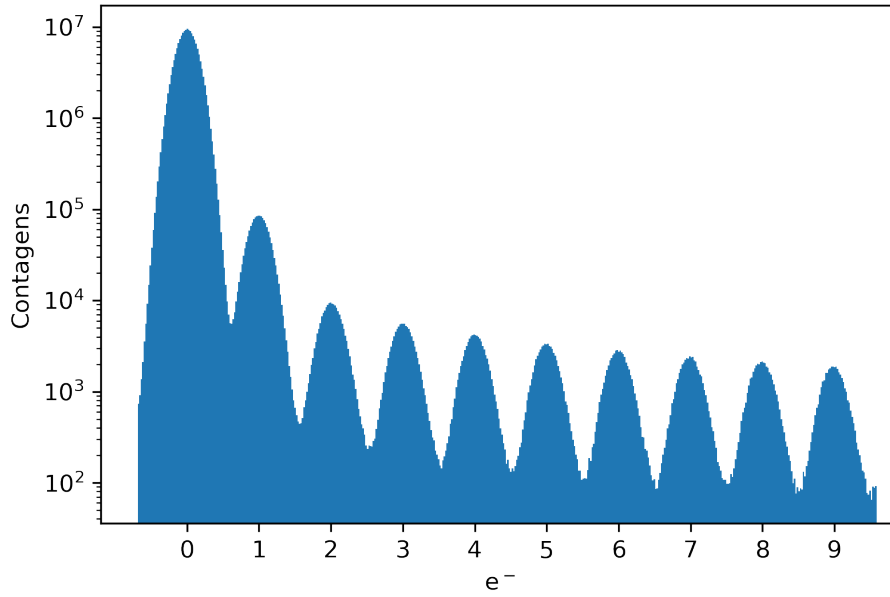


Figura 2.8 – Distribuição de dados de um conjunto de imagens produzidas por Skipper-CCD. As imagens foram somadas para aumentar a estatística de contagem em cada um dos picos.

## 2.5 DIFUSÃO DAS CARGAS

Depois que são criados no volume ativo do sensor, os portadores de carga são arrastados até os poços potenciais, onde ficam armazenados. Neste percurso, os pacotes de carga sofrem um processo de difusão e são desviados pelo campo elétrico aplicado, se alocando em pixels adjacentes, o que é ilustrado na figura 2.9, que mostra as trajetórias de cada portador com linhas tracejadas. A dispersão dos portadores é proporcional ao tempo que eles levam até chegarem ao poço, na superfície do pixel, por isso cargas criadas em regiões profundas se difundem mais. Isto é evidenciado claramente nos traços dos múons, apresentados nas figuras 2.4 e 2.5, onde podemos observar que a largura perpendicular à trajetória varia ao longo dela.



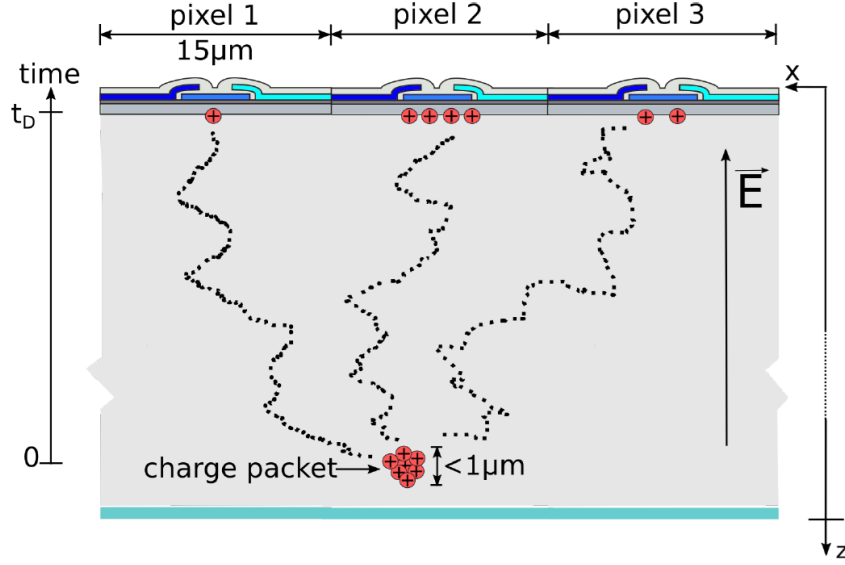


Figura 2.9 – Os portadores do pacote de cargas são difundidos pelo campo elétrico  $E$  até os poços potenciais na superfície dos pixels. A dispersão é proporcional ao tempo  $t_0$ . Figura extraída de [33].

Para modelar a relação entre a largura do padrão de difusão ( $\sigma_{xy}$ ) e a profundidade da ionização ( $z$ ), foi realizado um estudo com traços de múons, uma vez que essas partículas atravessam toda a região ativa dos detectores produzindo sinais alongados e retilíneos. Esses eventos apresentam uma extremidade mais espessa, associada à ionização produzida no fundo do sensor, e uma extremidade mais fina, correspondente à ionização próxima ao poço de potencial do pixel. A largura do traço pode ser mapeada em função da distância ao ponto inicial de interação, que é proporcional à profundidade do evento no CCD (figura 2.10 à esquerda). Na prática, o traço é dividido em segmentos, e em cada um deles se mede o desvio padrão transversal. Este mapeamento, realizado com muitos múons, está exemplificado na figura 2.10 à direita.

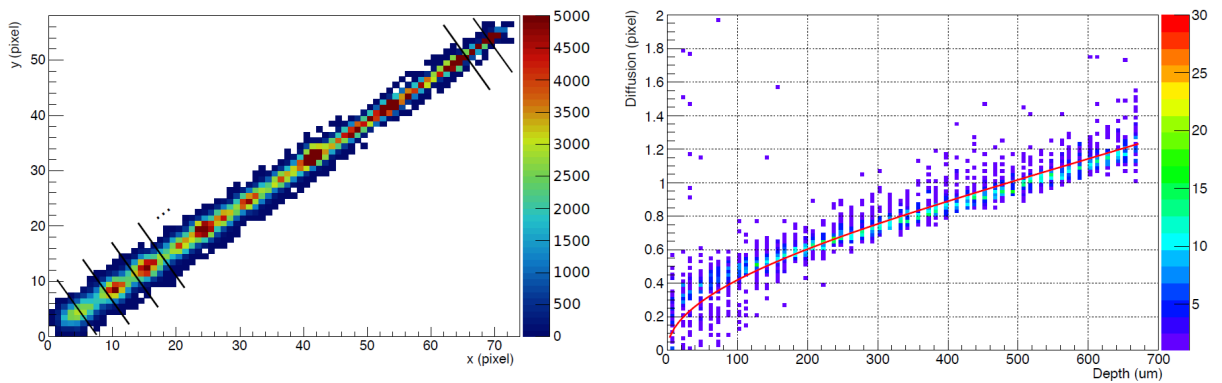


Figura 2.10 – Traço de um múon que atravessa toda o volume do CCD (à esquerda) e curva de calibração largura-profundidade (à direita). Figura retirada de [34].

A curva ajustada sobre a distribuição é representada pela equação 2.4, onde os parâmetros  $\alpha$  e  $\beta$  sintetizam as propriedades físicas do detector e podem variar.

$$\sigma_{xy} = \frac{\sqrt{\alpha \ln(1 - \beta z)}}{15} \quad (2.4)$$

Neste trabalho, esta fórmula foi implementada com  $\alpha = -205.157 \mu m^2$  e  $\beta = 1.052 \times 10^{-3}/\mu m$ , que são os valores obtidos para os sensores que operam no CONNIE (capítulo 3) com  $675 \mu m$  de espessura. O valor do quociente na equação 2.4 representa o tamanho de cada pixel, igual a  $15 \mu m$  para os sensores que estamos utilizando. Desta forma, a equação 2.4 vai nos fornecer a largura do padrão de difusão em unidades de pixels.

Através da calibração largura-profundidade podemos identificar os eventos candidatos ao espalhamento antineutrino-núcleo. Enquanto as interações de fundo que produzem eventos de carga pontual tendem a ocorrer nas regiões mais superficiais do sensor, os antineutrinos podem interagir em toda a sua espessura, originando também eventos em camadas mais profundas. Assim, a profundidade se torna um parâmetro essencial para distingui-los. Interações de baixas energias, como  $CE\nu NS$ , produzem eventos de carga com um padrão de dispersão pontual que pode ser ajustado por uma distribuição gaussiana bidimensional, cujo desvio  $\sigma_{xy}$  representa a largura do padrão de difusão, como pode ser observado na figura 2.11.

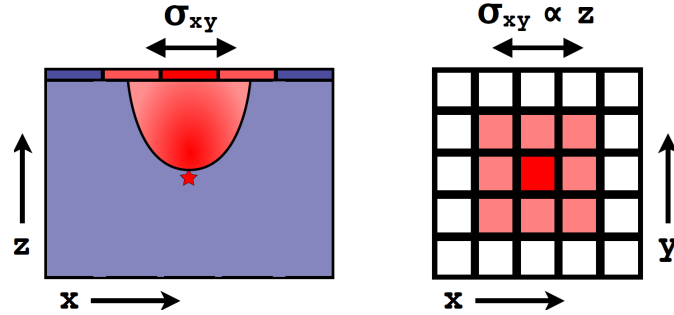


Figura 2.11 – Difusão de um pacote de carga criado em uma profundidade  $z$  (à esquerda), que produz o padrão de difusão com largura igual a  $\sigma_{xy}$  (à direita). Figura extraída de [35].

O algoritmo implementado para determinar a profundidade ( $z$ ) dos eventos de carga pontual faz a estimativa da largura do padrão de difusão ( $\sigma_{xy}$ ) através das projeções unidimensionais de carga nos eixos  $x$  e  $y$ . Para exemplificar este processo vamos utilizar uma imagem simulada com o algoritmo que desenvolvi (figura 2.12) e que será detalhado mais adiante, no capítulo 4. Esta imagem de  $20 \times 20$  pixels foi simulada contendo um único evento de carga pontual, com energia sorteada dentro do intervalo de 250 a 400 eV e gerado a uma profundidade igual a  $600 \mu m$ . A primeira etapa do algoritmo consiste em fazer as projeções da distribuição das cargas do evento nos eixos  $x$  e  $y$ . Depois, os histogramas destas distribuições são ajustados por funções gaussianas para obter os parâmetros  $\sigma_x$  e  $\sigma_y$ .

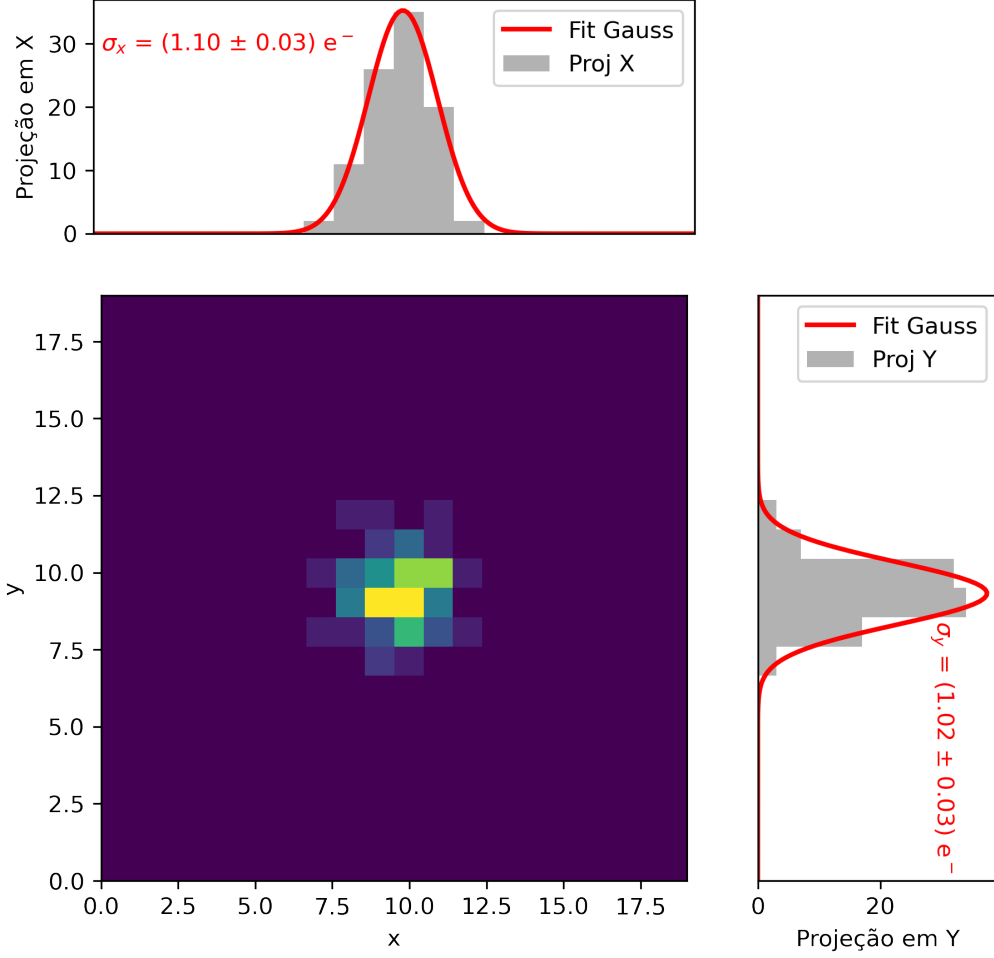


Figura 2.12 – Imagem com único evento de carga pontual simulado. Acima temos a projeção da distribuição de dados no eixo x e na lateral direita temos a projeção no eixo y. Um ajuste gaussiano é feito sobre cada projeção a fim de obter o desvio padrão, que representa a largura.

O parâmetro  $\sigma_{xy}$  é determinado a partir de  $\sigma_x$  e  $\sigma_y$ , assumindo que os eventos são pontuais e, por consequência, simétricos nos eixos:

$$\sigma_{xy} = \frac{\sqrt{\sigma_x^2 + \sigma_y^2}}{2} \quad (2.5)$$

Apesar de apresentar uma forma prática e, relativamente simples, de se obter o parâmetro  $\sigma_{xy}$ , esta abordagem tem limitações, especialmente nos casos em que os eventos de carga pontual possuem poucos pixels. Com o número baixo de *bins* no histograma da distribuição de dados, os ajustes nas projeções x e y são comprometidos, o que torna a estimativa de  $\sigma_{xy}$  imprecisa. A proposta a ser apresentada neste trabalho consiste no desenvolvimento de um algoritmo baseado em aprendizagem de máquina que seja capaz de prever, com maior precisão e acurácia, a profundidade de eventos de carga pontual. Para inferir este parâmetro a partir das imagens, o método adequado é uma rede neural convolucional, que extrai características e estruturas locais por meio da aplicação de filtros

convolucionais sobre as imagens. Diferentemente do algoritmo de CONNIE, as redes neurais vão prever a profundidade da interação diretamente do padrão de difusão dos eventos de carga pontual, sem intermédio da estimativa de um parâmetro  $\sigma_{xy}$ . Este tipo de ferramenta possui muito mais graus de liberdade e flexibilidade do que uma regressão tradicional, o que lhe confere capacidade de reconhecer padrões complexos nas imagens. Para serem treinadas e validadas as rede neurais precisam de grandes conjuntos de dados. Através do algoritmo de simulação de imagens que desenvolvi (capítulo 4), podemos simular eventos de carga pontual com energia e profundidades variadas, o que torna factível a implementação de uma rede neural para a solução deste problema (capítulo 6).



## Capítulo 3

# Experimento CONNIE

O experimento CONNIE (Coherent Neutrino Nucleus Interaction Experiment) está instalado ao lado do domo de contenção da usina nuclear de Angra 2 (como pode ser observado na figura 3.1) a aproximadamente 30 metros do reator, que opera com 3.8 GW de potência térmica resultando em um fluxo de  $7.8 \times 10^{12} \bar{\nu}_e/\text{cm}^2/\text{s}$ . Ao longo de mais de uma década de operação, CONNIE tomou dados com diferentes configurações experimentais. A figura 3.2 mostra a evolução do aparato experimental no decorrer dos anos [36], e detalha os períodos de comissionamento, a janela de operação com CCDs (Charged Coupled Devices) científicos e a etapa de uso de Skipper-CCDs, que se estende até os dias atuais. Com a implementação de CCDs científicos e, mais recentemente, Skipper-CCDs em seu aparato, o experimento alcançou recordes mundiais no limiar de detecção, apresentando os melhores resultados de taxa de ruído de fundo para experimentos de Skipper-CCDs em superfície [37]. Neste capítulo serão abordados as configurações do arranjo experimental, bem como especificações sobre os sensores empregados até então e o modo como operam. Vamos discutir ainda a cadeia de aquisição e processamento dos dados e os últimos resultados do experimento, obtidos com Skipper-CCDs.



Figura 3.1 – À esquerda está representada a localização do experimento CONNIE, na Central Nuclear Almirante Álvaro Alberto. À direita temos indicado o contêiner, onde o aparato experimental está instalado, posicionado ao lado do domo de contenção do reator de Angra 2. Figura adaptada de <https://www.eletronuclear.gov.br/>.

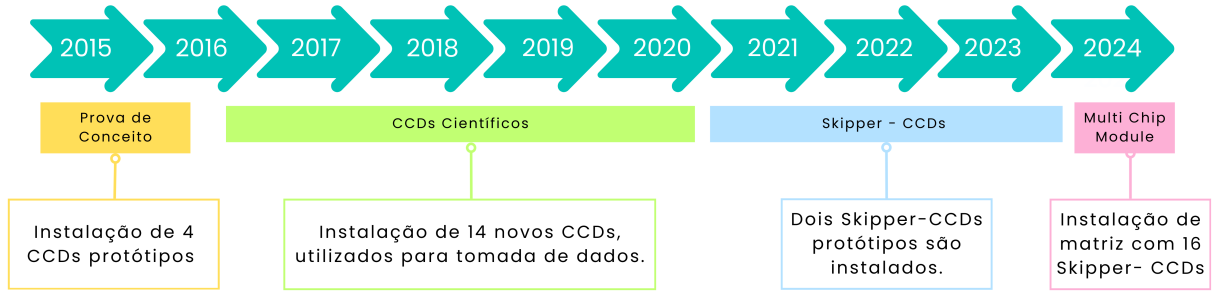


Figura 3.2 – Linha do tempo mostrando a evolução das diferentes fases e *setups* experimentais do CONNIE ao longo da última década.

### 3.1 ARRANJO EXPERIMENTAL

Em suas diferentes fases de operação, o experimento manteve um aparato comum de blindagem, resfriamento e suporte dos detectores. Os sensores são montados sobre um *wafer* de silício e conectados a um cabo flexível. Esta estrutura é fixada a uma moldura de cobre, que fornece suporte mecânico e facilita as conexões entre os sensores e a eletrônica de leitura (figura 3.3, à esquerda), isto compõe o que chamamos de 'empacotamento' dos sensores (*sensor packaging*). As bandejas com os detectores são então encaixadas, como uma espécie de gaveta, em uma caixa fria de cobre onde ficam empilhadas (figura 3.3, à direita). Para atenuar os efeitos da emissão térmica espontânea de portadores de carga (corrente escura) os sensores do experimento CONNIE operam em temperaturas criogênicas, da ordem de 100 K. Por isso, a caixa de cobre é posicionada dentro de uma câmara de vácuo (figura 3.4), a uma pressão de aproximadamente  $10^{-7}$  torr, evitando qualquer tipo de condensação dos sensores, e resfriada por um *cryocooler* de circuito fechado Gifford-McMahon [36]. Os cabos flexíveis dos diferentes sensores são conectados a uma placa VIB (Vacuum Interface Board), que atua como interface entre os sensores dentro da câmara de vácuo e o sistema eletrônico de leitura.

Como o experimento opera no nível do mar, está suscetível a contaminações devido ao alto fluxo de radiações cósmicas, e também provenientes do próprio ambiente, como por exemplo, os raios  $\gamma$ . Por conta disto, a câmara de vácuo é rodeada por uma blindagem passiva composta por duas camadas de 30 cm de polietileno de alta densidade intercaladas por uma camada de 15 cm de chumbo. A camada externa de polietileno protege os sensores de nêutrons cósmicos, o chumbo, por sua vez, blindas os sensores de raios  $\gamma$ . A camada mais interna de polietileno serve para proteção contra os nêutrons que são produzidos nas interações dos múons no chumbo. Até então, o experimento operou com 4 arranjos de detecção distintos: CCDs protótipos dedicados a prova de conceito, depois substituídos por CCDs científicos; posteriormente Skipper-CCDs protótipos; e, por último, um módulo integrado com 16 Skipper-CCDs (figura 3.2).



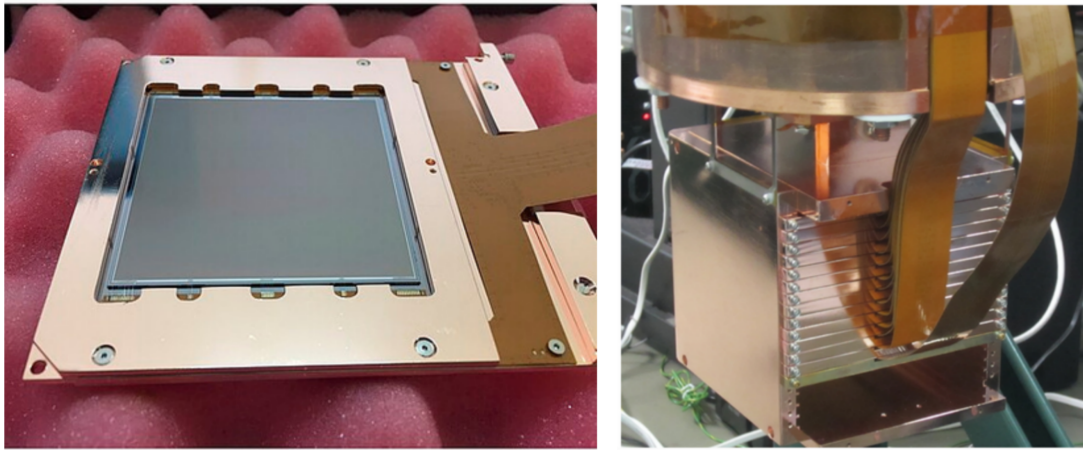


Figura 3.3 – À esquerda temos o sensor CCD de 16 megapixels montado sobre a moldura de cobre e conectado a um cabo flexível. À direita, a caixa de cobre onde estes sensores são posicionados em pilha. Figuras extraídas de [34].

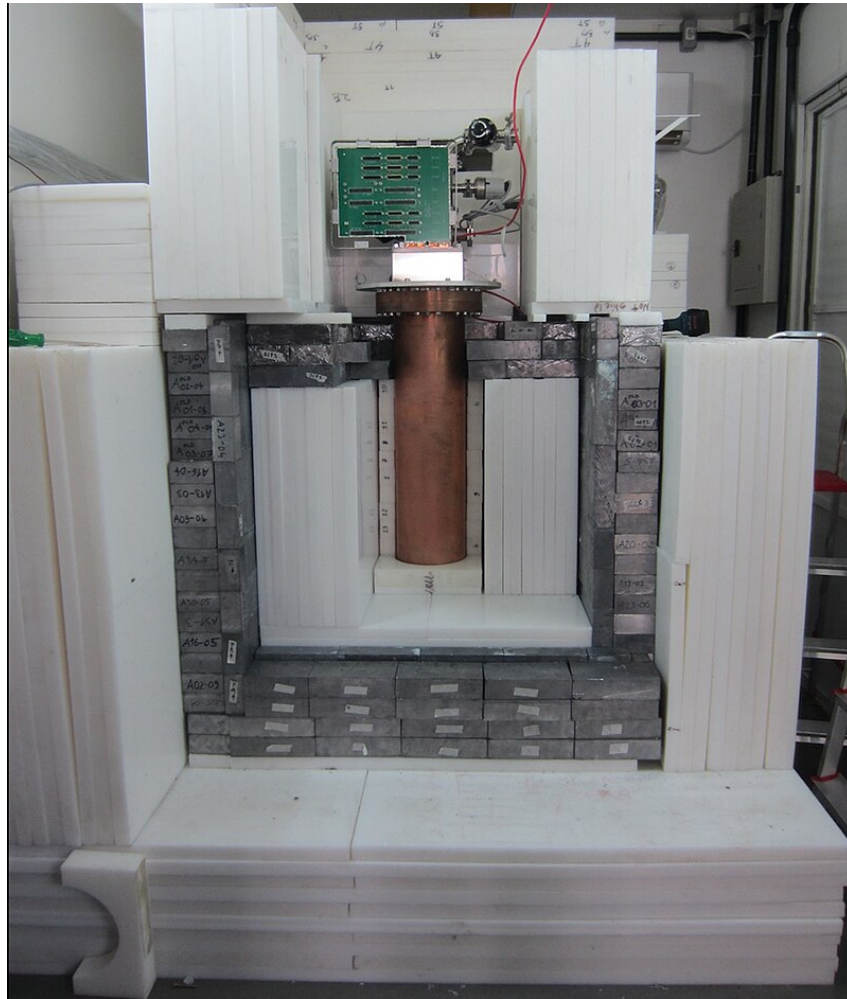


Figura 3.4 – Blindagem passiva do experimento composta por duas camadas de polietileno com 30 cm de espessura e uma de chumbo, de 15 cm. Ao centro temos a câmara de vácuo onde os sensores são dispostos. Figura extraída de [34].



### 3.1.1 CCDs

Os sensores CCDs instalados no experimento CONNIE foram fabricados em silício altamente resistivo ( $18 \text{ k}\Omega \text{ cm}$ ) numa parceria conjunta do Fermilab com os laboratórios Lawrence Berkeley National Laboratory Micro Systems (LBNL) [38]. Eles possuem dimensões de  $4\text{k} \times 4\text{k}$  pixels, com pixels de área igual a  $15 \times 15 \mu\text{m}$  e  $675 \mu\text{m}$  de espessura. Os detectores são conectados ao sistema eletrônico de controle e leitura Monsoon [39], que fica fora da caixa fria, através dos cabos flexíveis e da VIB. Os CCDs eram expostos durante 3 horas para logo serem lidos de forma sequencial, de acordo com o processo mencionado na seção 2.2, por um amplificador posicionado em um dos cantos. Este tempo de exposição foi definido a fim de garantir uma taxa de ocupação dos pixels abaixo dos 10%. Ao mesmo tempo, o segundo amplificador instrumentado lia uma imagem sem cargas para registrar as flutuações devido ao ruído correlacionado, que seriam posteriormente subtraídas durante o processamento de imagens [34]. Desta forma, a cada 3 horas uma imagem de todo o CCD era guardada para posterior análise de dados. Assim, imagens com ruído de leitura entre  $1.5 \text{ e}^-$  e  $2 \text{ e}^-$  eram obtidas. Informações mais detalhadas do uso de CCDs no experimento CONNIE podem ser encontradas em [34, 40].

### 3.1.2 SKIPPER-CCDs

Em meados de 2021, foram instalados no experimento CONNIE dois sensores Skipper-CCDs. Assim como os CCDs convencionais, eles foram produzidos em silício de alta resistividade desenhados pelo LBNL. Cada um dos sensores possui  $1022 \times 682$  pixels, totalizando uma área de  $1.57 \text{ cm}^2$  com 0.5 gramas de massa (figura 3.5). Eles foram posicionados nas regiões inferiores da caixa de cobre. A eletrônica de leitura utilizada para os Skipper-CCDs foi substituída por placas LTAs (Low-Threshold-Acquisition readout boards) [41], que permitem operar os detectores com máxima performance. A aquisição dos dados é concebida em duas etapas, a primeira consiste na inversão da voltagem para reduzir as contribuições de corrente escura e liberar qualquer outro tipo de carga acumulada no sensor. As tensões são retornadas aos valores usuais de operação e a limpeza dos detectores é concluída depois de ler e descartar as cargas da parte ativa consecutivas vezes. Devido à leitura recursiva das cargas de cada pixel, os Skipper-CCDs não podem ter o mesmo tipo de exposição que os CCDs comuns, já que os longos períodos de leitura resultariam em um sensor totalmente povoado por eventos de *background* (principalmente múons). Por este motivo, nestes sensores, o tempo de exposição de cada pixel é não uniforme. O primeiro pixel dos Skipper-CCDs possui tempo de exposição nulo, pois é o primeiro a ser lido depois da limpeza, enquanto o último pixel possui máxima exposição, que corresponde ao tempo que o sensor leva para ler todos os pixels anteriores.

Cada Skipper-CCD possui 4 amplificadores posicionados em cada um dos cantos, no entanto, para ter a melhor performance com a maior exposição possível, um único

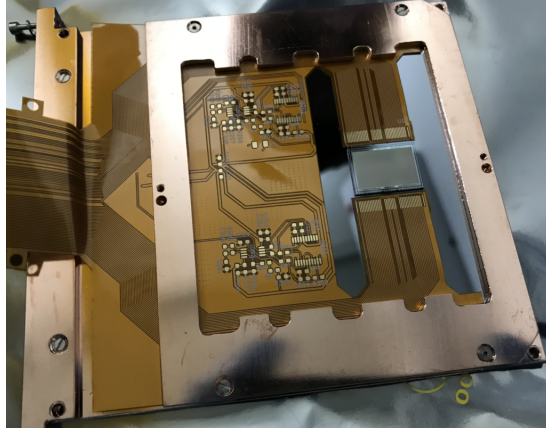


Figura 3.5 – Skipper-CCD de 1022 x 682 pixels montado sobre um substrato de silício e acoplado a um suporte de cobre junto a um cabo flexível. Figura extraída de [37].

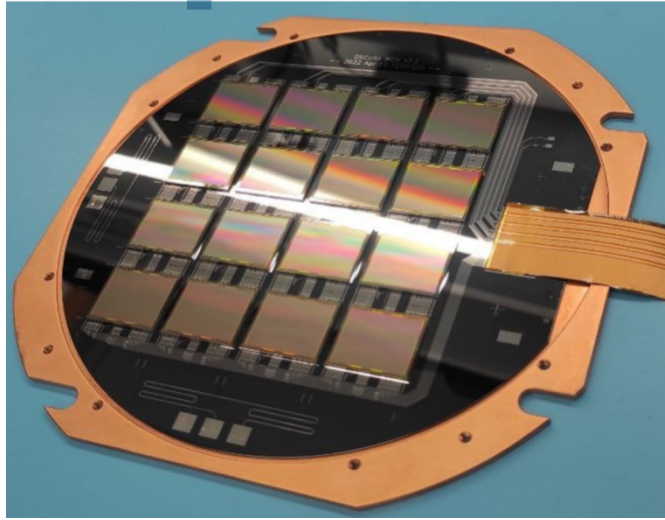


Figura 3.6 – Arranjo de 16 sensores Skipper-CCDs montados sobre um suporte e conectado a um cabo flexível para leitura. Figura extraída de [42].

amplificador é utilizado para ler metade dos detectores formando imagens de 420 x 1022 pixels, incluindo o OS. Mais detalhes sobre este modo de operação podem ser encontrados em [21]. A leitura dos sensores foi feita utilizando 400 amostras por pixel, o que resulta na média de 2 horas para construir uma imagem bruta. A tomada de dados é conduzida em rodadas de cerca de 500 imagens que compartilham as mesmas condições de operação, estes conjuntos são chamados de *runs*. Nos diferentes *runs* gerados o ruído de leitura oscilou de 0.135 até 0.170 elétrons, com um valor médio igual a 0.150 elétrons. Já a taxa de emissão de elétron único teve variações entre 0.010 e 0.120  $e^-/\text{px}/\text{dia}$ , com um valor médio de 0.045  $e^-/\text{px}/\text{dia}$ . O processamento destas imagens e a metodologia de calibração implementada serão discutidos na seção 3.2.

### 3.1.3 MULTI-CHIP-MODULE (MCM)

Em sua atualização mais recente, o experimento CONNIE instalou um módulo com 16 sensores Skipper-CCDs, chamado de MCM (Multi-Chip-Module), que é mostrado na figura 3.6. Estes sensores foram desenvolvidos para o experimento OSCURA [43], que busca empregar esta tecnologia para a detecção direta de matéria escura. Cada Skipper-CCD possui 1278 x 1058 pixels, e somados resultam em um módulo de 8 gramas, o que representa um aumento de massa em um fator 16 quando comparado a um dos Skipper-CCDs instalados anteriormente. Eles são dispostos como uma matriz de 4 x 4 sensores, fixados a um substrato de silício conectado a um cabo flexível. Eles foram idealizados para operar com sistema de leitura multiplexado, o que permite que os sinais produzidos nos 16 sensores sejam lidos por uma única placa LTA de eletrônica. Esta nova arquitetura de detecção está atualmente em fase de comissionamento.

## 3.2 PROCESSAMENTO DE DADOS

Nos anos iniciais do funcionamento de CONNIE com CCDs tradicionais, a rotina de processamento de dados era diferente da empregada atualmente para os Skipper-CCDs. Esta cadeia antiga é abordada detalhadamente em [21]. Nesta seção vamos nos dedicar ao novo algoritmo desenvolvido para imagens adquiridas com sensores Skipper-CCDs.

As imagens são geradas no formato FITS e são constituídas por três regiões: *prescan*, região ativa e *overscan*, como mostrado na figura 3.7. O *prescan* é formado a partir da leitura de pixels vazios que existem no registro serial, e antecede à leitura dos pixels que formam a parte ativa. A parte ativa é onde estão concentradas as interações físicas produzidas no volume do detector. Já o *overscan* (OS) corresponde à leitura de pixels vazios após a transferência de todas as cargas acumuladas no registro serial para o amplificador. Essa região é usada para avaliar o ruído de leitura e para corrigir flutuações da linha de base ao longo do processo de leitura.

Essas flutuações da linha de base decorrem do próprio processo de movimentação das cargas até o amplificador, podendo introduzir deslocamentos sistemáticos na distribuição de valores. Para corrigi-las, calcula-se a mediana dos pixels na região de OS e subtrai-se essa contribuição linha a linha. A figura 3.8 mostra, em sequência, uma imagem bruta, a estimativa da componente de linha de base e o resultado após a correção. Este processamento de imagens é feito automaticamente para cada *run*.

O baixo ruído de leitura dos Skipper-CCDs resulta em imagens de altíssima resolução, o que evidencia pequenos defeitos nos pixels e anomalias na difusão das cargas. Estas contribuições emulam sinais de carga pontual e por isso precisam ser caracterizadas e excluídas da fase de clusterização. As duas principais fontes de fundo são os eventos de registro serial manifestos nas imagens como deposições contínuas de carga ao longo



Figura 3.7 – Imagem de Skipper-CCD com as três regiões. A região delimitada à esquerda é o prescan (formada por 8 colunas), no centro temos a parte ativa (que nesta imagem possui 341 colunas) e à direita o overscan (formada por 70 colunas).

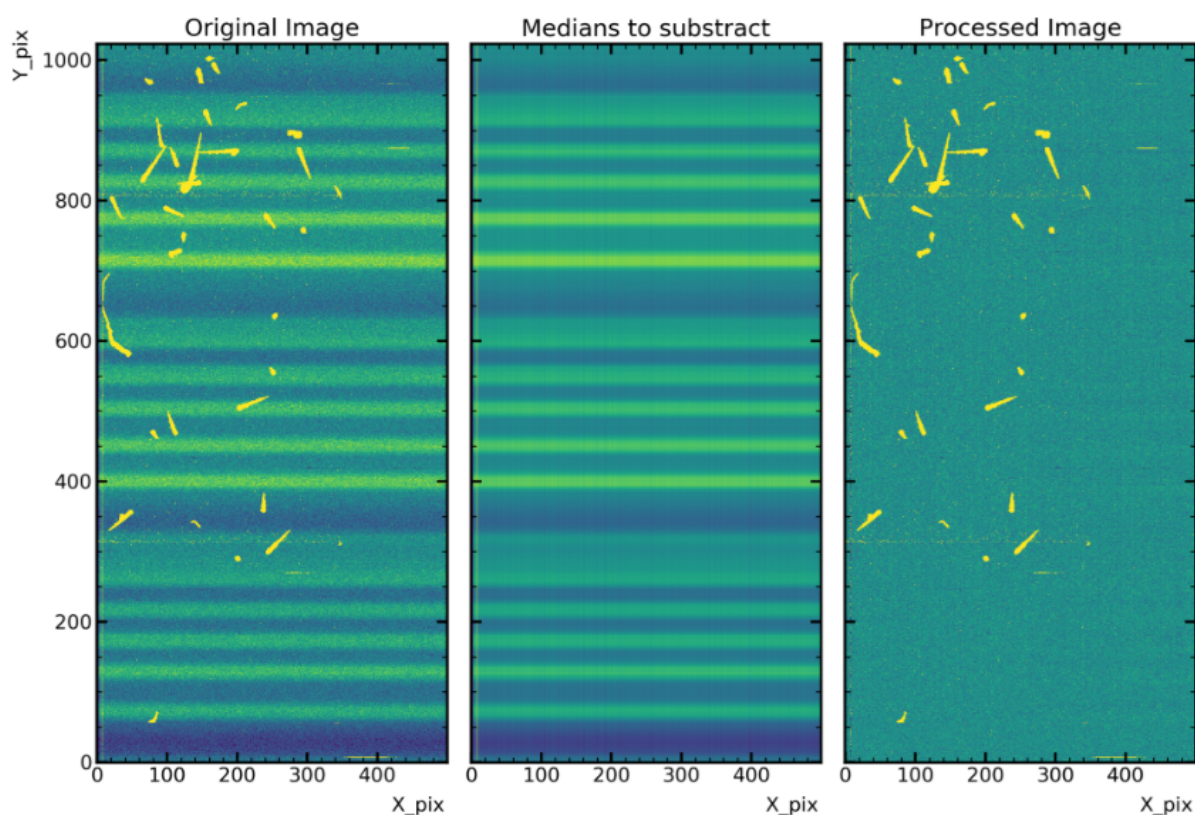


Figura 3.8 – Imagem bruta (à esquerda), linhas de base a serem subtraídas (centro) e imagem processada, já com a subtração (à direita). Figura extraída de [21].

de uma única linha e os pixels quentes, responsáveis pela produção de linhas e colunas saturadas. Dois processos de mascaramento distintos foram desenvolvidos para lidar com estes artefatos, e a junção dos dois dá origem a uma máscara global. Discussões adicionais sobre este procedimento podem ser consultadas em [21].



### 3.3 CALIBRAÇÃO DAS IMAGENS

Na etapa de leitura, a carga de cada pixel é medida e um valor de tensão proporcional é atribuído a este pixel no nó de detecção (seção 2.2). Este sinal é amplificado e segue para o conversor analógico digital, onde é quantificado em unidades digitais arbitrárias (ADUs). Para que sejam convertidos e interpretados em unidades físicas precisamos realizar um processo de calibração. O método de calibração que desenvolvi é baseado no fato de que os Skipper-CCDs têm resolução de elétron (o que fica claro observando a distribuição de carga da figura 2.8), ou seja, é possível identificar o valor, em ADUs, a que correspondem cada um dos picos. Primeiro é feita uma calibração imagem a imagem com o intuito de acompanhar a estabilidade do ganho e excluir imagens que ficam fora do comportamento global esperado. A evolução do ganho local para imagens de um mesmo *run* é mostrada na figura 3.9. Para cada imagem duas funções gaussianas são ajustadas aos picos de 0 e 1 elétron. O ganho ( $\text{ADU/e}^-$ ) é definido pela diferença das médias obtidas no ajuste. As imagens boas, que são aquelas que possuem ganho com desvio relativo ao valor médio menor que 5%, são utilizadas em uma calibração global.

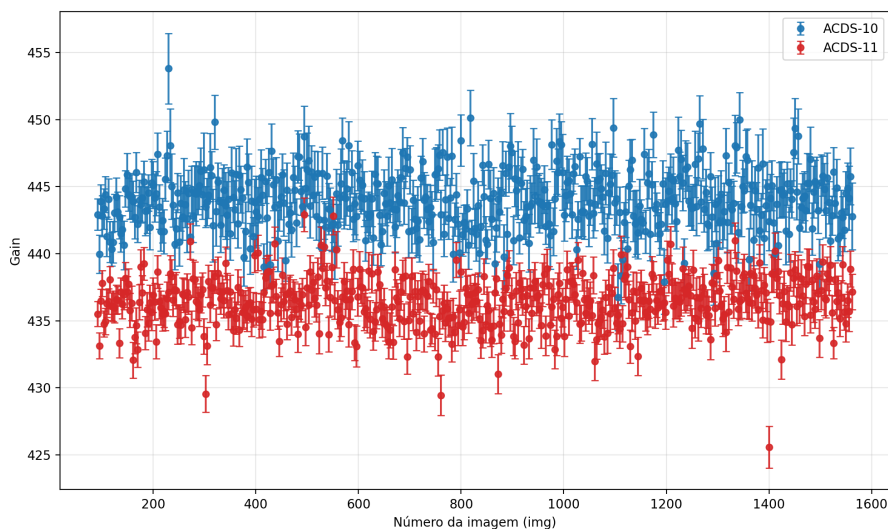


Figura 3.9 – Evolução da constante de calibração ao longo das imagens de um mesmo *run* para os dois Skipper-CCDs (ACDS-10 e ACDS-11).

Na segunda etapa, faremos o histograma de distribuição de carga de todas as imagens de um mesmo *run* que tenham sido classificadas como boas, e ajustamos funções gaussianas independentes aos 100 primeiros picos da distribuição total, como representado na figura 3.10(a). Por fim, o ganho é determinado por um ajuste linear feito sobre as médias das gaussianas ajustadas em função do número de elétrons, como mostra a figura 3.10(b).

Para estudar a linearidade da metodologia desenvolvida, utilizamos métricas de não linearidade próprias para conversores analógico-digitais. A não linearidade diferencial (DNL) é uma porcentagem calculada para cada par de picos adjacentes da distribuição de dados através da seguinte fórmula:

$$DNL_k = \left( \frac{\mu_{k+1} - \mu_k}{g} - 1 \right) \times 100, \quad (3.1)$$

onde,  $k$  é o número de elétrons do pico,  $\mu$  é a estimativa para a média dos picos em ADUs e  $g$  é o ganho em ADUs/e<sup>-</sup>. Para a metodologia de calibração desenvolvida, a DNL apresentou valores inferiores a 1%, como mostra a figura 3.11(a).

Também utilizamos a não linearidade integral, que é dada para cada pico da distribuição, em ADUs, pela fórmula:

$$INL_k = \mu_k - \mu_{ek}, \quad (3.2)$$

onde, mais uma vez  $\mu_k$  é a média para do pico e  $\mu_{ek}$  é o valor esperado para o pico obtido através do ganho. Os valores de INL obtidos para a metodologia em questão foram inferiores a 7%, como reportado pela figura 3.11(b).

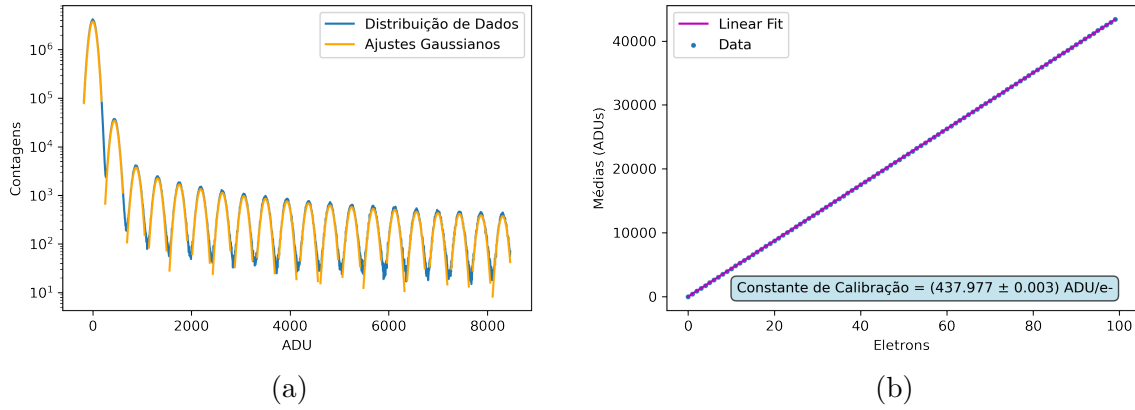


Figura 3.10 – Exemplo de ajustes gaussianos independentes aos primeiros 20/100 picos da distribuição de dados total (a). Ajuste linear sobre a relação das médias dos 100 primeiros picos em função do número de elétrons (b).

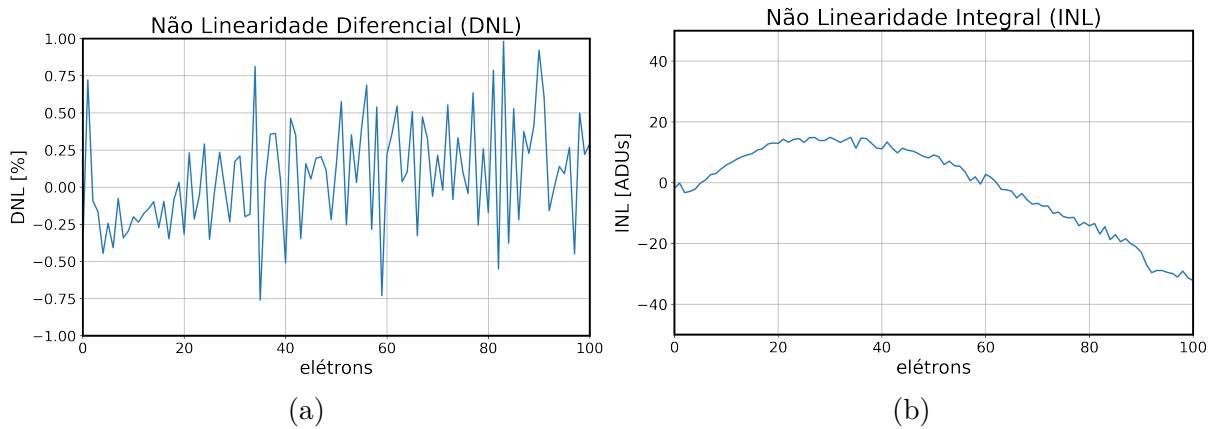


Figura 3.11 – Evolução das métricas DNL (a) e INL (b) em função do número de elétrons, para o regime de baixas energias.

Esta metodologia de calibração foi incorporada à cadeia de processamento das imagens de Skipper-CCDs do CONNIE criando um novo conjunto de imagens FITS, cujos valores de carga se encontram em unidades de elétron. Estas imagens são utilizadas pela colaboração para a análise de dados em conjunto com suas respectivas máscaras, também geradas durante esta cadeia de processamento.

### 3.4 IDENTIFICAÇÃO DE EVENTOS E ANÁLISE DE DADOS

A partir das imagens calibradas com suas respectivas máscaras a identificação e extração dos eventos é realizada, como detalhado na sessão 2.3. Neste processo os diferentes tipos de eventos são catalogados levando em conta se eles estão numa região mascarada ou não. Posteriormente, a adição de uma etiqueta torna possível selecionar apenas eventos nas regiões boas do sensor. Com base nos diferentes observáveis associados aos eventos, selecionamos os candidatos a  $\text{CE}\nu\text{NS}$  impondo sobre eles um corte em energia e um corte fiducial que nos permite eliminar os eventos de superfície com o intuito de reduzir o *background*. Uma vez que desenvolvemos estes cortes de seleção, podemos seguir para um estudo da eficiência de detecção em função da energia.

Este estudo foi feito simulando eventos de neutrinos (eventos de carga pontual) nas imagens obtidas durante o período de reator desligado. Para cada imagem foram sorteados 40 eventos de carga pontual num intervalo de energias de 5 a 2015 eV. Estas novas imagens são processadas da mesma forma que as imagens originais. Logo, a eficiência é determinada como o quociente entre o número de eventos simulados que passam nos cortes de seleção e o número total de eventos. Na figura 3.12 mostramos este quociente em função da energia reconstruída para os diferentes conjuntos de dados adquiridos com sensores CCDs e Skipper-CCDs. Nesta figura podemos ver que a eficiência satura em  $\sim 75\%$ , mas que no caso dos Skipper-CCDs, o limiar de detecção é significativamente mais baixo que no caso dos CCDs normais. Este resultado representa um grande avanço quanto à sensibilidade e é o melhor já obtido para um experimento que busca pelo espalhamento coerente antineutrino-núcleo.

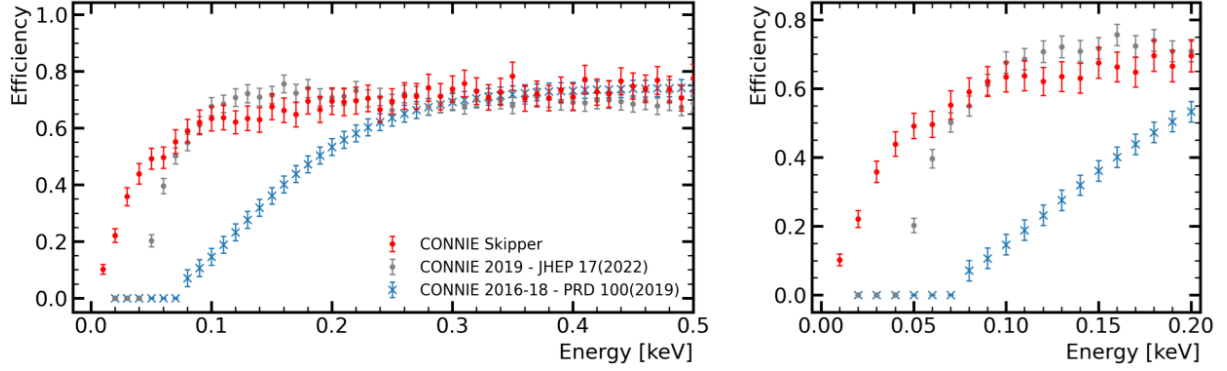


Figura 3.12 – Eficiência de detecção dos Skipper-CCDs do CONNIE (vermelho), considerando a aceitação na extração de eventos e os cortes de seleção. As eficiências obtidas para os conjuntos de dados de CONNIE com CCDs de 2016-2018 (azul) e de 2019 (cinza) são apresentadas como referência. À esquerda: Eficiência para energias de até 0.5 keV e à direita: um zoom na região de baixas energias. Figura extraída de [37].

Com os eventos que satisfazem os critérios de seleção estabelecidos construímos o espectro de energia para os períodos do reator desligado (*background*) e reator ligado. O conjunto de dados analisados resulta em uma exposição total de 18 g-dias, sendo 14.9 g-dias proveniente dos *runs* obtidos com o reator ligado e 3.5 g-dias com o reator desligado. Os espectros construídos para os dois períodos podem ser observados na figura 3.13, onde a unidade dru corresponde a eventos/kg/dia/keV. A diferença dos espectros não revela nenhum excesso significativo, o que é reportado na parte inferior da figura 3.13. No entanto, as baixas taxas de ruído de fundo motivam o uso do conjunto de dados para a busca de eventos raros fundamentados em física além do Modelo Padrão.



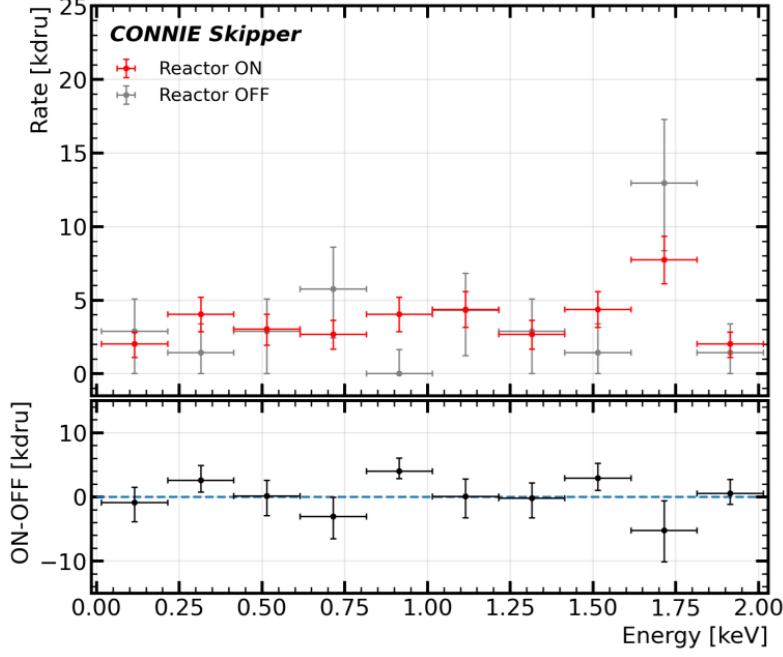


Figura 3.13 – Espectros obtidos nos períodos com reator ligado (em vermelho) e desligado (em cinza) e no bloco abaixo a diferença entre eles. Figura extraída de [37].

O impacto da redução de ruído alcançado com os Skipper-CCDs, aliado ao cuidadoso tratamento dos dados, pode ser avaliado a partir da comparação entre o espectro de *background* mostrado na figura 3.13 com os publicados em [34, 40] correspondentes a CCDs padrão. Um resultado particularmente relevante é a observação, pela primeira vez, de um espectro plano em baixas energias. A evolução do experimento, viabilizada pelo uso da tecnologia Skipper-CCD e pela estratégia de análise desenvolvida, levou a uma expressiva redução do excesso observado no primeiro bin de energia, onde a taxa diminuiu em cerca de 75%, atingindo valores estáveis da ordem de 4 kdru, compatíveis com os demais bins, que também apresentaram reduções de aproximadamente metade do valor em relação aos níveis anteriores.

### 3.5 RESULTADOS CIENTÍFICOS

Como discutido anteriormente, graças ao uso de Skipper-CCDs, a sensibilidade do experimento foi estendida para as mais baixas energias, com um limiar de detecção em 15 eV. Neste regime de baixas energias espera-se taxas mais altas de eventos devido à forma do espectro dos neutrinos do reator, como se pode ver na figura 1.1. A partir da diferença entre as taxas de eventos medidas com o reator ligado e desligado, mostrado na figura 3.13, estabelecemos um limite superior, com 95% de nível de confiança (C.L.) para a taxa de eventos observados em 76 vezes a predição do Modelo Padrão usando o fator de *quenching* de Sarkis [44]. Embora o limite atual resultante de  $\text{CE}\nu\text{NS}$  ainda não esteja próximo do previsto pelo Modelo Padrão, devido às incertezas estatísticas do

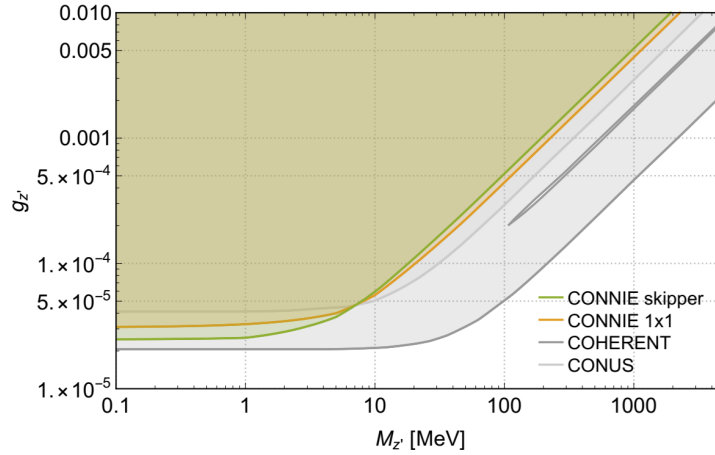


Figura 3.14 – Comparação entre os limites de exclusão para um mediador vetorial leve com 95% de nível de confiança (C.L.) obtidos pelo CONNIE utilizando Skipper-CCDs e CCDs padrão, e os obtidos pelos experimentos CONUS e COHERENT [46]. Figura extraída de [37].

pequeno conjunto de dados, ele é comparável ao atingido no nosso resultado anterior utilizando CCDs, que foi obtido com uma exposição duas ordens de magnitude maior [34, 40]. Portanto, os resultados demonstram uma sensibilidade aumentada para sondar  $\text{CE}\nu\text{NS}$  e novas interações de neutrinos em energias muito baixas. Várias interações de neutrinos não padrão (NSI, *Non-Standard Interactions*) previstas por extensões do Modelo Padrão podem ser investigadas a partir de  $\text{CE}\nu\text{NS}$ , uma vez que podem modificar a seção de choque do processo, especialmente em baixas energias. O experimento CONNIE, mesmo sem ter detectado  $\text{CE}\nu\text{NS}$ , colocou limites significativos em teorias além do Modelo Padrão, considerando um modelo com um mediador vetorial (escalar) leve  $Z'$  ( $\Phi$ ) com massa  $M_{Z'}$  ( $M_\Phi$ ), que foram os mais restritivos na época [22].

A partir dos resultados com Skipper-CCDs, colocamos novamente limites para modelos simplificados com mediadores vetoriais leves. O limite no espaço de fase do modelo representa uma melhoria em relação ao nosso resultado anterior, que foi obtido a partir de um conjunto de dados com exposição muito maior. O resultado é reportado na figura 3.14, onde se apresenta também uma comparação com outros resultados experimentais. Em virtude do alcance dos Skipper-CCDs em energias mais baixas, o limite é mais restritivo em massas baixas do mediador quando comparado ao experimento CONUS [45], que utiliza detectores de germânio.

Os Skipper-CCDs fornecem hoje as melhores restrições para matéria escura (DM) leve interagindo com elétrons, na faixa de 0.5 a 1000 MeV. Essas restrições são limitadas por eventos de um único elétron em baixas energias, mas podem ser aprimoradas pela busca de modulação diurna. O chamado "vento de DM", resultante do movimento do Sistema Solar na galáxia, produz uma direção preferencial de chegada das partículas. Isso gera uma modulação diária no fluxo de DM, que depende do "ângulo de isodeteção" entre essa direção e a superfície terrestre no detector. Enquanto experimentos no hemisfério norte

observam ângulos menores e, portanto, modulações fracas, no hemisfério sul a modulação é mais acentuada. O CONNIE, situado a  $23^\circ$  S, pode explorar ângulos de isotecção entre  $65^\circ$  e  $161^\circ$  [47].

Utilizando um subconjunto dos dados de CONNIE com Skipper-CCDs, onde a taxa de elétron único se manteve muito estável, realizamos um estudo de modulação diurna de matéria escura pela primeira vez. As taxas observadas foram comparadas a simulações com o DaMaSCUS [48], permitindo impor limites a 90% C.L. para modelos de DM na escala de MeV, que interagem via fóton escuro. Os resultados obtidos (figura 3.15) mostram limites para diferentes massas do mediador (pesado e ultraleve), com desempenho competitivo entre  $[0.6, 10]$  MeV, até 3 ordens de magnitude melhores que outros experimentos de superfície. Apesar de menos competitivos que os subterrâneos, devido ao fundo cósmico, os resultados são promissores para futuras buscas com maior estabilidade e melhor controle de fundos.

A sensibilidade nas escalas de eV alcançadas por CONNIE utilizando a tecnologia Skipper-CCD viabilizou o estudo das partículas de carga fracionária (mCP) [49]. Estas partículas propostas por várias extensões do SM poderiam ser geradas, em pares, por fótons de alta energia dentro de reatores nucleares e se manifestariam por interações electromagnéticas com a matéria, por meio do processo de ionização atômica. A seção de choque para o silício, recentemente calculada para a faixa dos eVs [50], revela um pico significativo de plasmon em torno de 15 eV, justamente próximo do limiar de detecção dos Skipper-CCDs. Isto coloca CONNIE em uma situação de privilégio em relação a outros experimentos existentes para estudar mCPs. Com os dados obtidos, fomos capazes de melhorar o limite de TEXONO [51] no valor da carga para a mCP no intervalo de massas de 1-106 eV. A partir da combinação dos resultados de CONNIE com os obtidos por Atucha [52], obtivemos o melhor limite mundial para mCP, como mostrado na figura 3.16. Estes são os primeiros resultados das análises de dados dos Skipper-CCDs por CONNIE e demonstram a robustez e versatilidade dos sensores, bem como seu potencial promissor para detectar  $\text{CE}\nu\text{NS}$  e sondar novos modelos de física com alta sensibilidade.

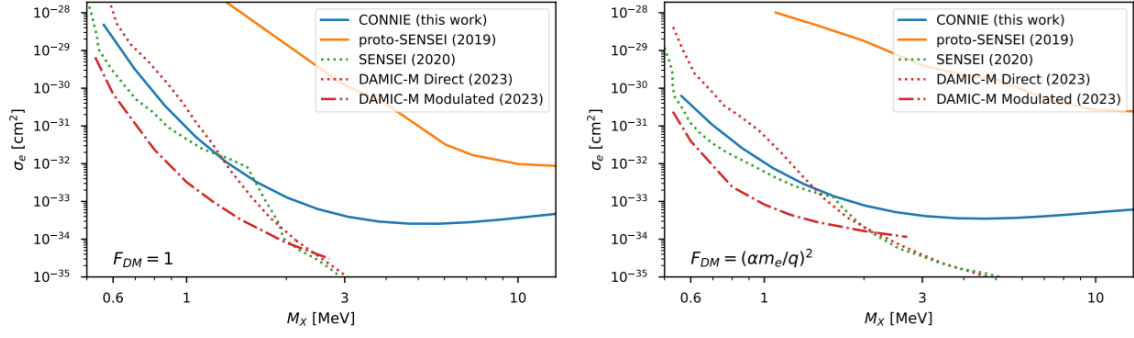


Figura 3.15 – Limites superiores com 90% de C.L. para interações matéria escura-elétron mediadas por um fóton escuro pesado (à esquerda) e um fóton escuro ultraleve (à direita), estabelecidos por CONNIE (azul, linha sólida), protoSENSEI [53] (laranja, linha sólida), SENSEI [54] (verde, linha pontilhada) e DAMIC-M [55, 56] (vermelho, linha tracejada para buscas com modulação e vermelho, linha pontilhada para buscas diretas). Linhas sólidas correspondem a experimentos realizados na superfície, linhas pontilhadas a limites diretos obtidos no subsolo, e linhas tracejadas a limites modulados obtidos no subsolo. Figura extraída de [37].

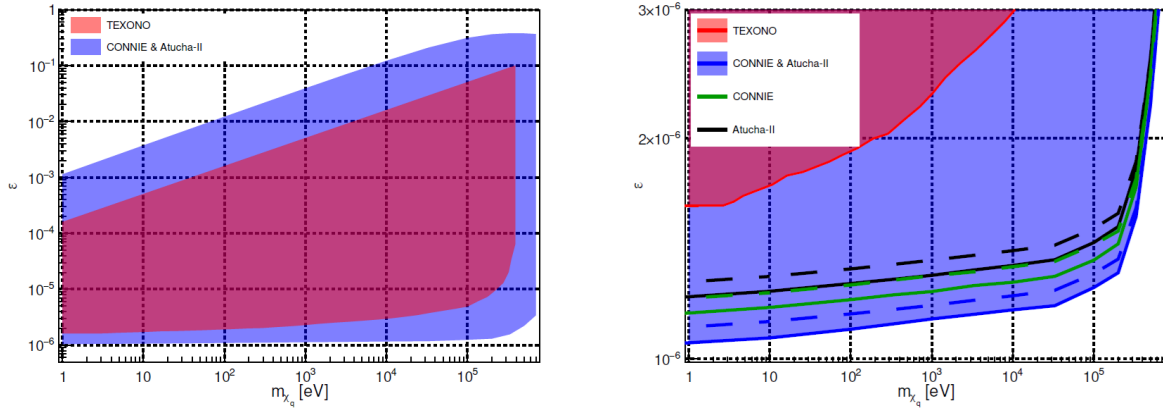


Figura 3.16 – Limites de exclusão a 90% de nível de confiança (C.L.) em função da massa e da fração de carga das partículas milicarregadas (mCP) para o CONNIE (verde), Atucha-II (preto) e sua combinação (azul). O limite de exclusão do TEXONO [51] também é incluído para comparação (vermelho). À esquerda: os limites superiores são obtidos a partir da atenuação das mCPs à medida que elas viajam do reator até o detector. À direita: linhas contínuas representam resultados considerando a produção de mCPs tanto por raios  $\gamma$  primários quanto secundários, enquanto linhas tracejadas consideram apenas os raios  $\gamma$  primários. Figura extraída de [49].



# Capítulo 4

## Simulação de Imagens

Um algoritmo de simulação de imagens foi desenvolvido com a finalidade criar um conjunto de eventos de carga pontual controlados que vou utilizar para treinar e validar redes neurais convolucionais. A simulação foi concebida como dois componentes: o plano de ruído de fundo das imagens e o plano das interações físicas, no nosso caso particular, as interações de carga pontual. As simulações foram realizadas através da linguagem *Python*, da qual utilizamos a biblioteca *astropy*, que nos permite gravar as imagens simuladas como arquivos do tipo *FITS*, o mesmo formato em que os dados do experimento CONNIE são guardados. Uma vez simuladas as imagens poderemos aplicar sobre elas todos os códigos de análise desenvolvidos para CONNIE e discutidos anteriormente, no capítulo 3. Aqui iremos detalhar como é feita cada etapa da simulação, como validamos a mesma e alguns resultados que foram obtidos além do objetivo deste trabalho, mas que contribuem com o entendimento da resposta dos sensores Skipper-CCDs ante diferentes performances.

### 4.1 SIMULAÇÃO DE IMAGENS VAZIAS

As imagens simuladas são compostas por um conjunto de pixels organizados em um arranjo bidimensional. Antes da simulação de qualquer interação física, é necessário simular para cada pixel uma carga associada ao plano de ruído de fundo resultante de fontes de ruído intrínsecas ao sistema de detecção. Sobre esse plano de fundo serão posteriormente adicionados os sinais dos eventos físicos de interesse. As principais fontes de ruído que compõem esse plano de fundo são o ruído de leitura e os eventos de elétron único, que podem ser produzidos por diferentes mecanismos, como corrente escura (dark current), cargas espúrias e luz proveniente do próprio amplificador, entre outros.

#### 4.1.1 RUÍDO DE LEITURA

O ruído de leitura em CCDs é dominado por contribuições associadas ao amplificador de saída MOSFET, responsável por converter a carga dos pixels em um sinal de tensão.

As duas fontes principais são o ruído branco térmico (Johnson–Nyquist), decorrente da agitação térmica dos portadores de carga nos resistores e no canal do MOSFET, e o ruído de *reset*, introduzido periodicamente no nó de detecção a cada ciclo de reinicialização. Ambos apresentam espectro aproximadamente branco na faixa de operação do CCD e dependem linearmente da temperatura, podendo ser descritos pela relação geral:

$$N_{W,R} = \sqrt{4kTB R_{W,R}}, \quad (4.1)$$

onde  $N_W$  e  $N_R$  são as tensões de ruído branco e ruído de *reset*,  $k$  é a constante de Boltzmann e  $T$  a temperatura absoluta, medida em K. A largura de banda de potência de ruído, medida em Hz, é denotada por  $B$ . Por fim,  $R_W$  e  $R_R$  são, respectivamente, as resistências do condutor e transistor.

Estes ruídos podem ser obtidos em unidades de elétrons se escrevermos a largura de banda de potência ( $B$ ), em função da capacitância ( $C$ ) através da fórmula:

$$B = \frac{1}{4RC}. \quad (4.2)$$

Substituindo a equação 4.2 na expressão 4.1, e convertendo a tensão para o número de elétrons por meio de  $C$  e da carga do elétron  $q$ , temos:

$$N_{R,W}(e^-) = \frac{\sqrt{kTC}}{q} \quad (4.3)$$

O efeito produzido pelo ruído de *reset* pode ser, contudo, bastante mitigado através da técnica de *Correlated Double Sampling*, na qual se mede o nível de saída imediatamente antes e depois do pulso de *reset*, para logo determinar o sinal como a diferença entre estes. Este processo cancela grande parte do ruído correlacionado ao *reset* de forma que a sua contribuição residual se torna desprezível em relação a outras fontes.

Desta forma podemos considerar que o ruído de leitura é dominado pelo ruído branco, sendo apropriado modelá-lo como uma distribuição gaussiana, dada por:

$$G(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}. \quad (4.4)$$

Nesta distribuição o ruído de leitura será representado pelo desvio padrão ( $\sigma$ ). Para simular o ruído de leitura das imagens recorreremos à função *random.gauss* que utiliza o algoritmo *Mersenne Twister* [57] como núcleo para gerar números pseudoaleatórios. O algoritmo leva este nome pois toma como base o número de *Mersenne* ( $2^p - 1$ ) para gerar as sequências numéricas determinísticas a partir de uma semente. A função recebe como argumento os valores da média e desvio padrão da distribuição. Assumimos que a distribuição estará sempre centrada em zero, já que este ruído não introduz deslocamentos sistemáticos na leitura das cargas, apenas oscilações em torno de seu valor nominal. Na figura 4.1 temos à esquerda, como exemplo, uma imagem de 70 x 140 pixels simulada apenas com ruído de

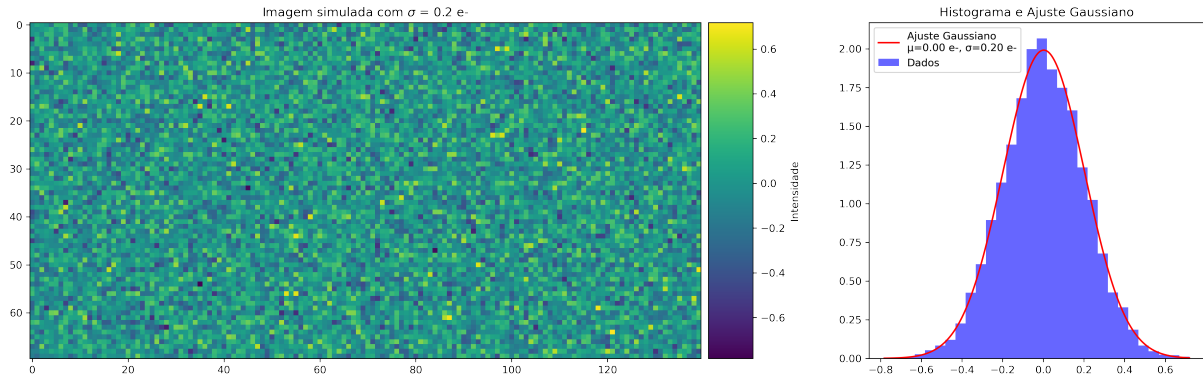


Figura 4.1 – Imagem de 70 x 140 pixels simulada com ruído de leitura de  $0.2 e^-$  (esquerda). Distribuição dos dados desta mesma imagem ajustada por uma função gaussiana (direita).

leitura igual a  $0.2 e^-$ ; à direita temos a distribuição de carga nos pixels onde podemos ver que recuperamos o ruído de leitura sorteado.

#### 4.1.2 EVENTOS DE ELÉTRON ÚNICO

A segunda contribuição é devida aos eventos de elétron único (SEE, *single electron event*). Entre estes, temos a geração de carga espúria, que ocorre durante a movimentação da carga no CCD, quando a variação dos potenciais nos eletrodos induz a geração de pares elétron-buraco mesmo na ausência de fótons incidentes. Além disto, há também a contribuição da luz do amplificador, responsável pela geração de elétrons únicos a partir da interação dessa radiação com o volume ativo do CCD. Estes eventos que se concentram nas proximidades do amplificador, são produzidos durante a etapa de leitura e sua taxa cresce linearmente com a duração do processo. Finalmente, temos a corrente escura (DC, *dark current*) que é intrínseca aos sensores semicondutores e consiste na emissão térmica espontânea de portadores minoritários, ocorrendo mesmo na ausência de interações de partículas externas. Para atenuar essa contribuição os sensores são operados a temperaturas criogênicas, tipicamente em torno de 140 K.

Como os eventos de elétron único são de baixa frequência, aleatórios e independentes entre si, sua ocorrência pode ser modelada por uma distribuição de Poisson, caracterizada por uma taxa média de eventos por unidade de tempo ( $\lambda$ ).

$$P(x) = \frac{(\lambda)^x e^{-\lambda}}{x!} . \quad (4.5)$$

Esta contribuição foi simulada através da função `np.random.poisson`, da biblioteca `NumPy`, que gera números pseudoaleatórios a partir de uma distribuição de *Poisson* e recebe o valor de  $\lambda$  como argumento. A unidade da taxa de emissão de elétron único é dada em  $e^-/\text{px}/\text{dia}$ . No entanto, para simular esta contribuição utilizamos a unidade de  $e^-/\text{px}/\text{img}$ . A conversão é feita através do tempo, em dias, necessário para se ler cada imagem. Por



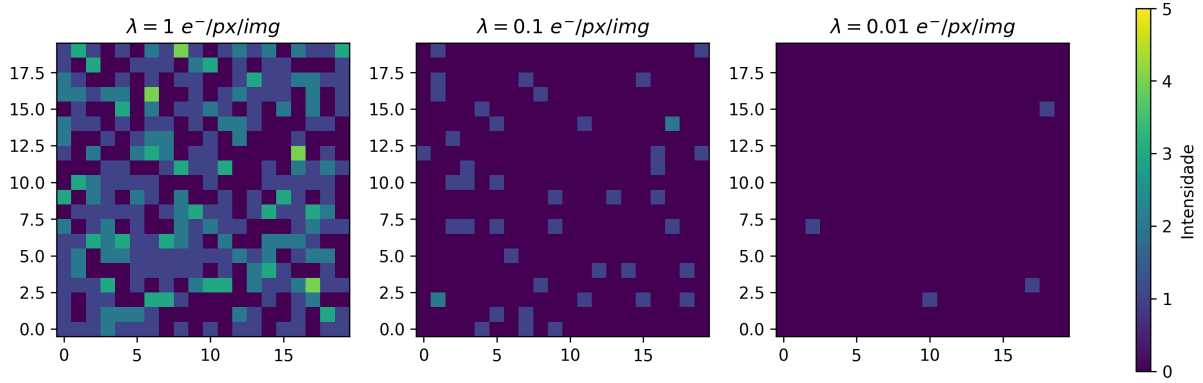


Figura 4.2 – Imagens simuladas apenas com eventos de elétron único com taxas de 1  $\text{e}^-/\text{px}/\text{img}$  (à direita), 0.1  $\text{e}^-/\text{px}/\text{img}$  (no centro) e 0.01  $\text{e}^-/\text{px}/\text{img}$  (à esquerda).

exemplo, se um dado sensor demora 2 horas para ler uma imagem e a taxa de emissão de elétron único associada é igual a 0.1  $\text{e}^-/\text{px}/\text{dia}$ , o valor desta taxa em termos do número de eletrons por pixel por imagem é obtido como  $8.33 \times 10^{-3} \text{ e}^-/\text{px}/\text{img}$ . A figura 4.2 mostra três imagens de 20 x 20 pixels, simuladas com diferentes taxas de emissão de elétron único. A primeira imagem, à esquerda, foi simulada com um  $\lambda$  de 1  $\text{e}^-/\text{px}/\text{img}$ , e é nítido como os eventos de elétron único dominam a imagem e se sobrepõe, gerando sinais de mais de 1  $\text{e}^-$ . No centro, na imagem simulada com 0.1  $\text{e}^-/\text{px}/\text{img}$ , temos uma quantidade menor de eventos de elétron único. E na imagem à esquerda, onde a taxa é de 0.01  $\text{e}^-/\text{px}/\text{img}$ , temos apenas quatro eventos de 1 elétron.

#### 4.1.3 SOMANDO AS DISTRIBUIÇÕES DE FONTE DE RUÍDO ASSOCIADAS AO PIXEL

Cada imagem é gerada pelo algoritmo como uma matriz  $N_y \times N_x$  (*array* bidimensional) de elementos nulos, onde  $N_x$  e  $N_y$  são o número de pixels nos eixos horizontal e vertical, respectivamente. Em seguida, simulamos as distribuições de ruído de leitura e corrente escura, com o número de elementos correspondentes ao número de pixels total da imagem. Estas distribuições são obtidas como um *array* unidimensional (*ndarray*), por isso precisam ser redimensionadas para que se tornem compatíveis com o formato da imagem. A última etapa consiste em somar as contribuições do ruído de leitura e da corrente escura elemento a elemento. A figura 4.3 mostra as três etapas deste procedimento para uma imagem de 20 x 20 pixels. No primeiro passo o ruído de leitura, de 0.2  $\text{e}^-$ , é simulado. No segundo, representado pela imagem ao centro, simulamos a taxa de emissão de elétron único com  $\lambda = 0.1 \text{ e}^-/\text{px}/\text{img}$ . Por último, as contribuições do ruído de leitura e da taxa de emissão de elétron único são somadas pixel a pixel, dando origem ao fundo de ruído onde serão adicionados os eventos.

A partir das imagens simuladas de plano de fundo, é possível recuperar os valores dos

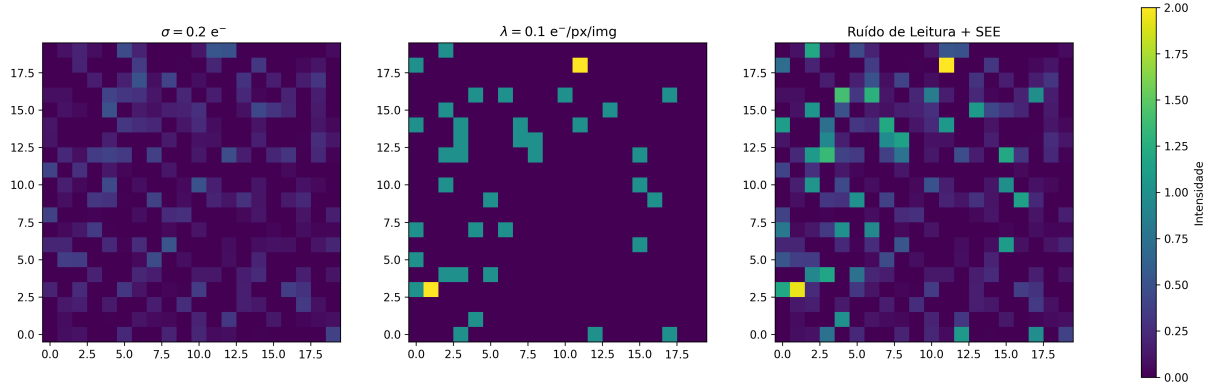


Figura 4.3 – Passos de simulação do ruído de fundo para uma imagem de 20 x 20 pixels: primeiro simulamos o ruído de leitura (à esquerda), depois a taxa de emissão de elétron único (centro) e, por fim, as duas contribuições são somadas na imagem final (à direita).

parâmetros  $\sigma$  e  $\lambda$  através do ajuste da função de convolução que é construída a partir das funções de Gauss e Poisson. Para exemplificar vamos usar uma imagem de 70 x 140 pixels, simulada com ruído de leitura de  $0.2 e^-$  e taxa de emissão de elétron único igual a  $0.1 e^-/\text{px}/\text{img}$ . A imagem simulada é mostrada na figura 4.4, à esquerda. A partir desta imagem, construímos um histograma de distribuição da carga em cada pixel, e ajustamos sobre ele a função de convolução para estimar os parâmetros  $\sigma$  e  $\lambda$  da distribuição, como mostra a figura 4.4, à direita.

Os parâmetros obtidos ajustando os primeiros 3 picos são:

$$\begin{aligned}\sigma &= (0.201 \pm 0.001) e^- \\ \lambda &= (0.094 \pm 0.005) e^- \text{ px}^{-1} \text{ img}^{-1}.\end{aligned}$$

Os resultados reproduzem os valores esperados a partir da simulação, validando assim a modelagem de fundo construída a partir das principais fontes de ruído intrínsecas ao sistema de detecção.

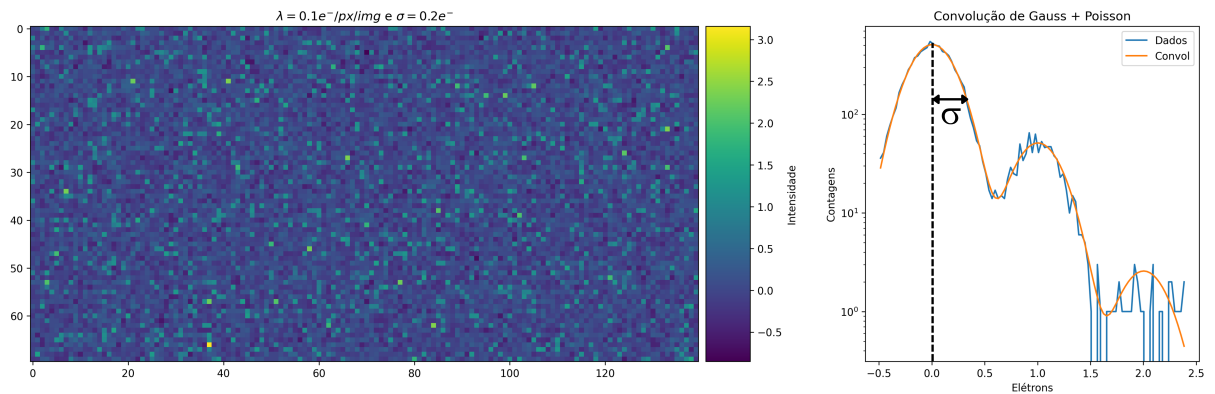


Figura 4.4 – Imagem simulada de tamanho 70 x 140 pixels com ruído de leitura igual  $0.2 e^-$  e taxa de emissão de elétron único de  $0.1 e^-/\text{px}/\text{img}$  (esquerda). Ajuste da convolução sobre a distribuição de dados (direita).

## 4.2 SIMULAÇÃO DOS EVENTOS DE CARGA PONTUAL

Os eventos de carga pontual são simulados como uma distribuição gaussiana bidimensional, que se difunde a partir de uma determinada profundidade. O número de eventos simulados por imagem é previamente definido. Já as variáveis de posição ( $x$ ,  $y$  e  $z$ ) são sorteadas através de uma distribuição uniforme, utilizando a função *random.uniform* do Python, já que estes eventos podem ser produzidos em qualquer parte do volume ativo do sensor. No caso da energia do evento, neste trabalho decidimos por, também simular eventos com distribuição uniforme, já que o espectro de energia observado para o *background* (sessão 3.4), nos dados experimentais é compatível com uma distribuição plana acima de 15 eV. A energia destes eventos é sorteada dentro de um intervalo, tipicamente definido entre 15 eV e 1015 eV. A partir deste valor determinamos o número médio de elétrons gerados na interação assumindo que:

$$N_{\text{medio}} = \frac{E}{E_I} , \quad (4.6)$$

onde  $E$  é a energia sorteada para a interação e  $E_I$  é a energia de ionização a 140 K necessária para produzir um par elétron-buraco no Silício.

O fator de Fano ( $F$ ) [29] representa as flutuações estatísticas no número de portadores de carga gerados e, no nosso algoritmo, ele é usado para corrigir o valor de  $N_{\text{medio}}$  tornando as simulações mais realísticas. Este fator depende das características do material, e para Skipper-CCDs de Silício possui valor igual a 0.12 [58]. A variância no número de portadores pode ser expressa como:

$$\sigma_F^2 = F N_{\text{medio}} . \quad (4.7)$$

A partir destes valores, o número de elétrons é sorteado com uma distribuição gaussiana centrada em  $N_{\text{medio}}$  e com desvio padrão igual a  $\sigma_F$ . Como o resultado do sorteio é um valor contínuo, tomamos a parte inteira desta variável.

O passo seguinte consiste em calcular a largura da difusão gaussiana ( $\sigma_{xy}$ ), o que é feito através da profundidade da interação  $z$  sorteada por meio da equação 2.4, discutida no capítulo 2. Para cada elétron gerado, sorteamos as coordenadas ( $x_{\text{random}}$  e  $y_{\text{random}}$ ) através de uma distribuição gaussiana centrada nas posições previamente sorteadas ( $x$  e  $y$ ) e com desvio padrão igual a  $\sigma_{xy}(z)$ , que vão sendo somados a uma imagem (array bidimensional) originalmente vazia. Na figura 4.5 observamos a sequência aqui descrita para o caso de um evento de energia de 38 eV que produz 10 elétrons. Finalmente mostramos na figura 4.6, uma imagem de 70 x 140 pixels simulada com 10 eventos de carga pontual com energias entre 500 eV a 1000 eV sobre um *background* típico com um ruído de leitura de 0.2 e<sup>-</sup> e uma taxa de eventos de elétron único de 0.1 e<sup>-</sup>/pix/img.

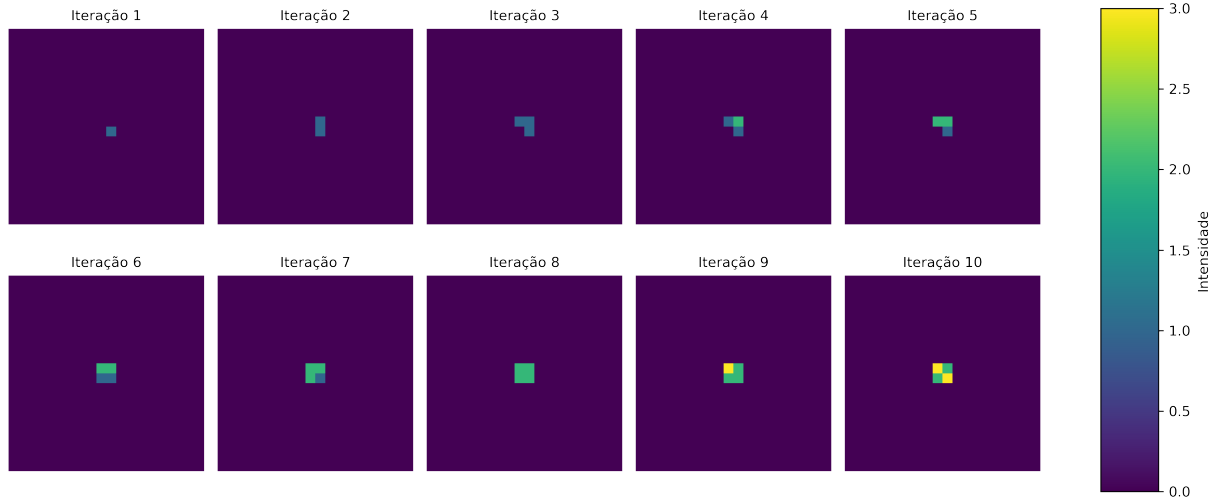


Figura 4.5 – Exemplo das iterações para sorteio de elétron em um evento de 38 eV, que produz 10 elétrons. Os elétrons são somados um a um ao longo das iterações.

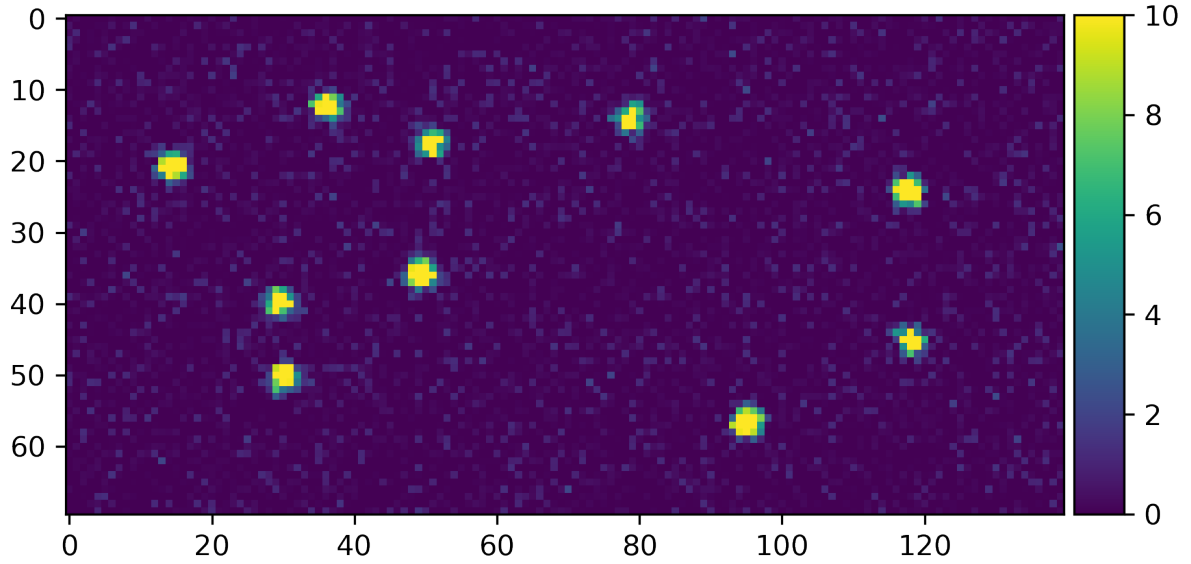


Figura 4.6 – Imagem simulada com  $\sigma = 0.2 \text{ e}^-$  e  $\lambda = 0.1 \text{ e}^-/\text{px}/\text{img}$ , com 10 eventos gerados em profundidade de  $675 \mu\text{m}$  e energia sorteada no intervalo de  $[500, 1000] \text{ eV}$ .

O algoritmo de simulações desenvolvido não tem utilidade apenas para o treinamento de redes neurais, como veremos mais adiante, no capítulo 6. Esta ferramenta também pode ser implementada para o estudo e validação do código de extração de eventos que faz parte da cadeia de análise de dados do CONNIE. Como os eventos de carga pontual são simulados com energias variadas, podemos fazer um estudo da eficiência e acurácia da reconstrução destes eventos em função de sua energia, por exemplo. Desta forma, o algoritmo contribui diretamente para o aprimoramento do processamento e análise de imagens empregado em CONNIE.

### 4.3 VALIDAÇÃO DO ALGORITMO DE SIMULAÇÃO

O algoritmo desenvolvido foi validado mediante a simulação de imagens similares às obtidas com Skipper-CCDs no CONNIE. Estas imagens foram reconstruídas utilizando a cadeia completa de análise de dados discutida na seção 3.4. Para isto, simulamos 2300 imagens de 1022 x 341 pixels com o plano de ruído de fundo parametrizado por  $\sigma = 0.16 \text{ e}^-$  e  $\lambda = 0.1 \text{ e}^-/\text{px}/\text{dia}$  (que corresponde a  $0.00645 \text{ e}^-/\text{px}/\text{img}$ ). Nestas imagens foram simulados eventos de carga pontual com energia sorteada dentro do intervalo de 15 até 1015 eV, com o número de eventos por imagem compatível com os 4 kdru que obtemos na análise de reator ON-OFF, que discutimos na seção 3.4. Os estudos de validação do algoritmo foram conduzidos com base neste conjunto de imagens simuladas.

#### 4.3.1 CARACTERIZAÇÃO DAS SIMULAÇÕES

Como explicado anteriormente, as imagens simuladas foram processadas pela mesma cadeia de análise que utilizamos para os dados do CONNIE. A partir disto, construímos um catálogo com todos os eventos extraídos de acordo com o procedimento explicado na seção 2.3. Na figura 4.7 mostramos um gráfico da posição dos eventos mediante as coordenadas do baricentro, assim como para suas projeções. Como podemos ver, as projeções seguem distribuições planas dentro das flutuações estatísticas esperadas.

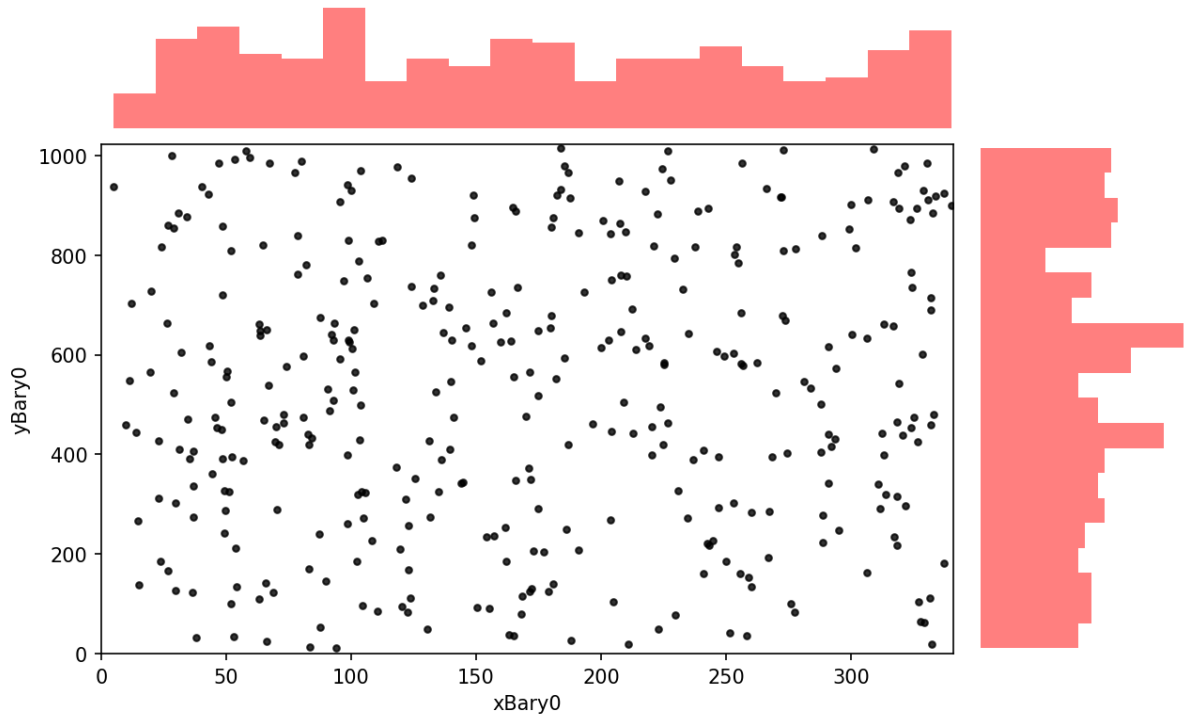


Figura 4.7 – Distribuições das projeções em x e y dos baricentros dos eventos extraídos e catalogados. Os histogramas, acima e à direita, representam as projeções dos baricentros dos eventos nos eixos x e y, respectivamente.

Também verificamos se a distribuição de energia dos eventos extraídos tem comportamento condizente com o previsto. Para isto, construímos um histograma da distribuição de energia dos eventos, em eV. Na figura 4.8 observa-se um excesso no primeiro bin de energia, em valores menores que 15 eV (limiar de detecção do experimento). Para entender de forma minuciosa este comportamento em baixas energias, fizemos um segundo histograma, que foi posicionado no canto superior direito da figura, para energias menores que 20 eV. O excesso possui energias dentro do intervalo de 8 eV até 10 eV. Isto aponta para a reconstrução de eventos falso positivos a partir de flutuações do ruído nos sensores, já que os eventos simulados nas imagens foram sorteados com energias maiores que 15 eV. Um estudo mais aprofundado sobre a reconstrução de eventos falso positivos em função do ruído de leitura e da taxa de emissão de elétron único pode ser encontrado no apêndice.

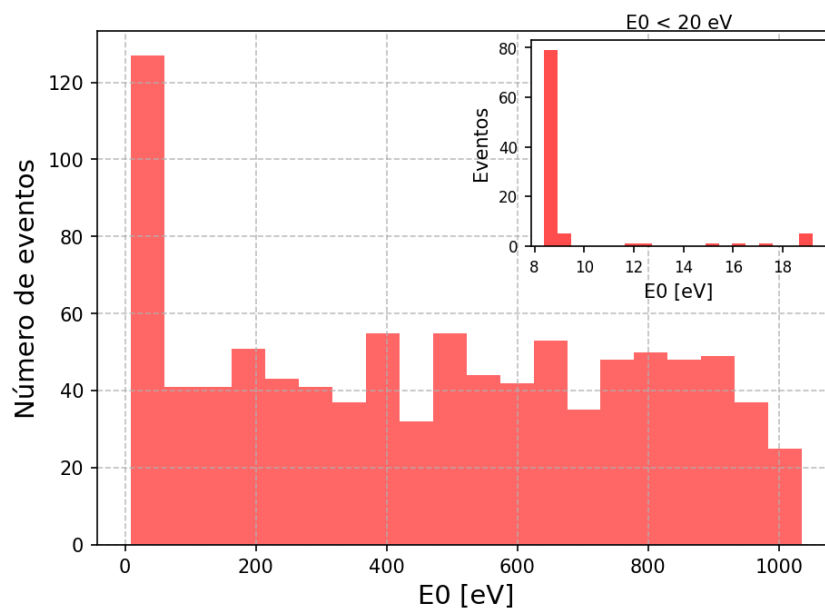


Figura 4.8 – Distribuição das energias reconstruídas para os eventos simulados. O histograma possui excesso no primeiro *bin*, o que condiz com a extração de eventos falso-positivos, representada pelo gráfico do canto superior direito.

Para os eventos acima de 15 eV, podemos comparar a energia reconstruída com a simulada, para entender a qualidade do processo de extração-reconstrução. Isto é feito através do gráfico da discrepância relativa entre os valores de energia simulados e reconstruídos. O *profile* desta diferença em função da energia simulada (figura 4.9) mostra que no regime de 200 a 1000 eV, a reconstrução dos eventos é bem sucedida com respeito a energia. Já para energias menores que 200 eV nota-se que as discrepâncias são maiores, o que indica que a reconstrução não é feita de forma acurada. Isto se deve ao fato de que eventos de baixas energias costumam englobar poucos pixels, o que pode comprometer a eficiência do processo de extração-reconstrução.

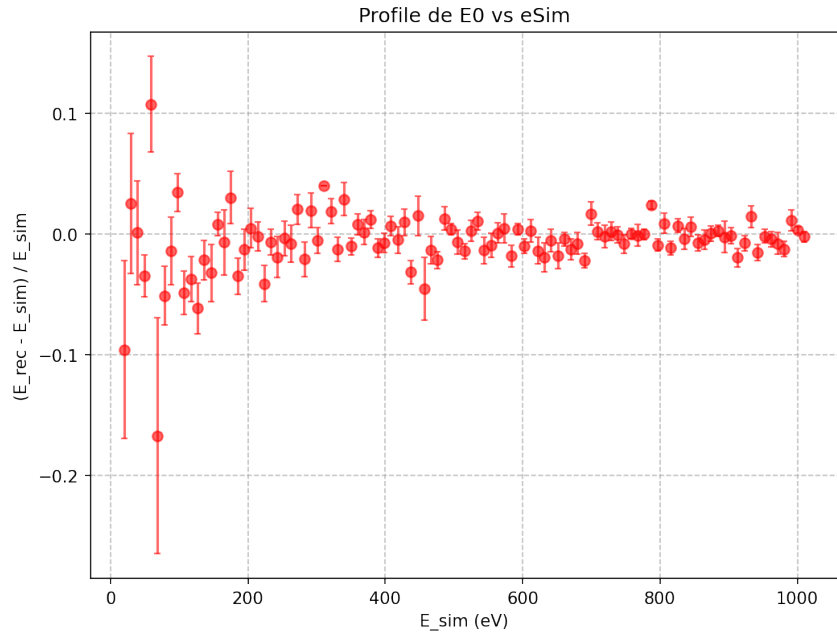


Figura 4.9 – *Profile* da discrepância relativa entre as energias simulada e reconstruída.

### 4.3.2 COMPARAÇÃO DE ESPECTROS

Para comparar os espectros das imagens simuladas com o espectro real, produzido pelo experimento CONNIE para o caso de reator OFF, construímos um histograma com 5 *bins* de larguras iguais a 0.2 keV. O valor das contagens de cada *bin* são divididos pela massa dos sensores simulados, em kg, pelo tempo de leitura correspondente, em dias, e pela largura do *bin*, em keV. Desta forma, obtemos as taxas de eventos em unidade de dru (eventos/kg/dia/keV).

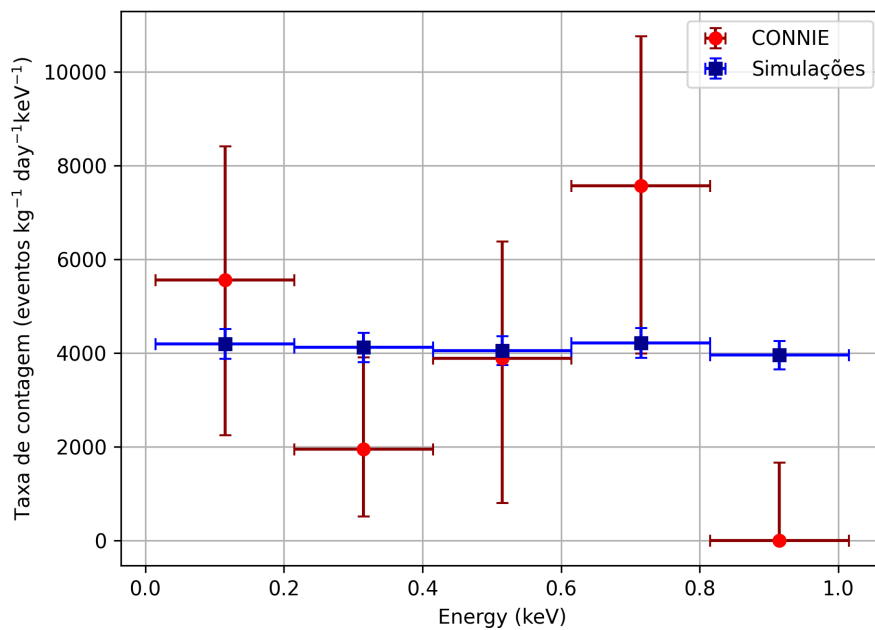


Figura 4.10 – Sobreposição dos espectros real e simulado.

A comparação entre os dois espectros está reportada na figura 4.10. As poucas estatísticas obtidas no experimento CONNIE são refletidas pelas maiores barras de erro, em comparação com as do espectro simulado. No geral, há um acordo global entre os dois espectros.





# Capítulo 5

## Redes Neurais Convolucionais

### 5.1 INTRODUÇÃO

A inteligência artificial é um campo das ciências computacionais que se dedica a desenvolver sistemas computacionais com capacidades analíticas similares às de um ser humano, de maneira que possam executar atividades complexas. Com o uso deste conjunto de técnicas, as máquinas são capazes de fazer reconhecimento de imagem e voz, identificar padrões, fazer classificações e previsões, por exemplo. Este tema ganhou proeminência popular com o surgimento dos modelos de inteligência artificial generativas, que criam textos e imagens com base em instruções humanas, como é o caso do *ChatGPT* e do *DALL-E*, programas desenvolvidos pela *OpenAI*. No entanto, este domínio da computação não é recente, mas já vem sendo estudado e aprimorado há mais de 80 anos.

Um dos subconjuntos da inteligência artificial é o aprendizado de máquina, onde os modelos aprendem a partir dos dados. Há, essencialmente, dois tipos de aprendizagem automatizada: o caso supervisionado, onde os exemplos são acompanhados de rótulos e o algoritmo faz o mapeamento das entradas de acordo com a saída estipulada; e o não supervisionado, onde o modelo busca reconhecer padrões nos dados de entrada. Dentro da aprendizagem de máquina, temos o aprendizado profundo, que emprega redes neurais artificiais com várias camadas para ensinar máquinas a reconhecerem padrões ou fazerem previsões de forma automatizada. As redes neurais convolucionais (CNNs) são um segmento do aprendizado profundo (conforme ilustrado no diagrama da figura 5.1) e são caracterizadas pelo uso de filtros convolucionais em suas camadas ocultas, o que as tornam ferramentas poderosas para o processamento de imagens. Ao longo das seções seguintes vamos descrever como uma rede neural convolucional é formada e seu funcionamento.

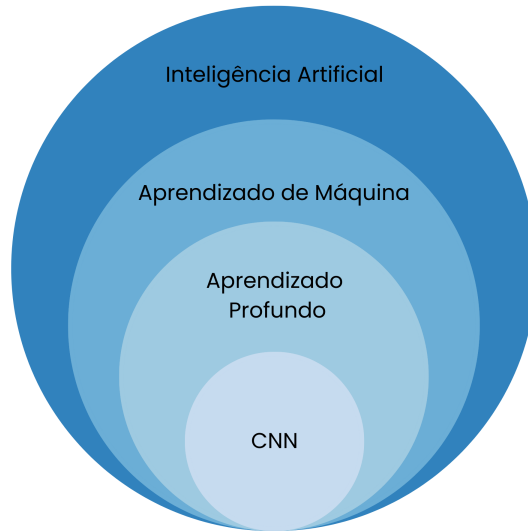


Figura 5.1 – Diagrama ilustrativo dos subconjuntos da inteligência artificial.

A primeira contribuição estruturante para o conceito de inteligência artificial foi dada em 1943, por Warren McCulloch e Walter Pitts [59], com a proposta de descrever a atividade neural por meio de representações lógicas. O neurônio de McCulloch-Pitts foi a primeira modelagem matemática de um neurônio biológico. Os neurônios são as células básicas do cérebro humano, estes componentes são responsáveis pela recepção e transmissão de informações. Devido à sua excitabilidade, os sinais propagados por células neurais vizinhas causam alterações elétricas na membrana do neurônio receptor. Se a soma das excitações recebidas ultrapassar um determinado limiar crítico o neurônio dispara um impulso elétrico, que é transmitido por toda sua extensão.

No artigo de McCulloch e Pitts, a atividade neural é caracterizada como 'tudo ou nada', o que permite que ela seja representada por uma lógica binária. O neurônio artificial idealizado por eles recebe um conjunto de entradas ( $x = x_1, \dots, x_n$ ), a cada uma destas entradas é associado um ganho sináptico ou peso ( $w = w_1, \dots, w_n$ ), que pode ter caráter excitatório ( $w_i > 0$ ) ou inibitório ( $w_i < 0$ ). Para determinar a saída do neurônio é necessário calcular a média ponderada das entradas com seus respectivos pesos:

$$y = \sum_{i=1}^n w_i x_i \quad (5.1)$$

Se a contribuição dos estímulos ( $y$ ) for suficiente para superar o limiar de excitação ( $\theta$ ) o estado correspondente é 1 e há transmissão de um impulso nervoso. No caso contrário, o estado é 0. Para comparar estas grandezas e definir a saída do neurônio utiliza-se uma função de ativação do tipo 'degrau'.

$$f(y) = \begin{cases} 1, & \text{se } y \geq \theta \\ 0, & \text{se } y < \theta. \end{cases} \quad (5.2)$$

Este dispositivo matemático foi útil para entender operações booleanas e representar circuitos lógicos simples. Mas o fato de ter seus pesos e limiares previamente fixados era

impeditivo para lidar com problemas complexos. Anos mais tarde, com a proposta de aprimorá-lo, Frank Rosenblatt desenvolveu um algoritmo neural mais sofisticado conhecido como *perceptron* [60]. Diferentemente do neurônio de McCulloch-Pitts, o *perceptron* foi concebido com a capacidade de aprender com os dados disponibilizados e fazer o ajuste dos parâmetros automaticamente a partir do erro entre a saída prevista e a saída correta. O processo de aprendizagem de um *perceptron* pode ser representado com a seguinte sequência de passos:

- *Passo 1*

O algoritmo recebe um vetor de entradas que representa as características de um observável. Este vetor é associado a um rótulo que indica a saída esperada.

$$x = (x_1, x_2, \dots, x_n) \xrightarrow{\text{rótulo}} y$$

- *Passo 2*

Cada elemento do vetor de entrada é multiplicado pelos pesos.

$$x_i w_i$$

- *Passo 3*

A soma das entradas aplicadas aos pesos é feita com a adição de um *bias*,  $w_0$ .

$$\sum_{i=1}^n x_i w_i + w_0$$

- *Passo 4*

A este somatório é aplicada uma função de ativação  $\varphi$ , que retorna o valor de saída.

$$\varphi \left( \sum_{i=1}^n x_i w_i + w_0 \right) = \bar{y}$$

- *Passo 5*

O valor de saída é comparado com o rótulo e os pesos são atualizados a fim de minimizar o erro  $(y - \bar{y})$ .

$$w_{i+1} = w_i + \eta(y - \bar{y})x_i, \quad \text{onde } \eta \text{ é a taxa de aprendizagem}$$

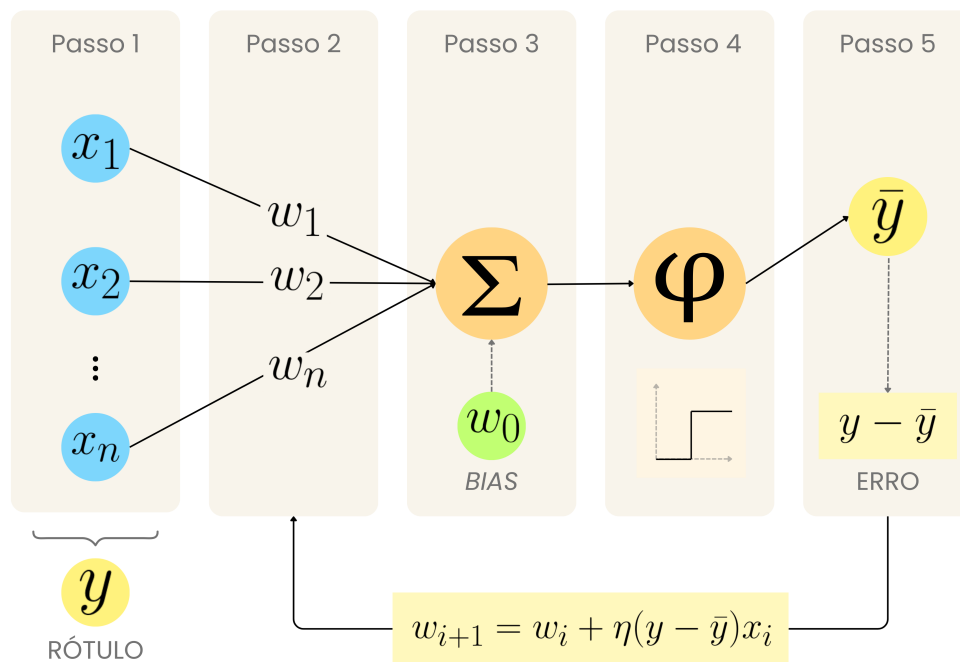


Figura 5.2 – Diagrama ilustrativo do processo de aprendizagem de um *perceptron* para um único vetor de entrada.

A figura 5.2 esquematiza o processo de aprendizagem de um *perceptron* para um único vetor de entrada. No entanto, para que os modelos aprendam com os exemplos de forma eficiente, é necessário fornecer inúmeros pares de vetores e rótulos. Desta forma o algoritmo será capaz de reconhecer padrões específicos nos dados e generalizar suas classificações ou previsões para novas amostras. O aprendizado é, essencialmente, o ajuste dos pesos na tentativa de minimizar a função de perda. Esta função quantifica a diferença entre os rótulos fornecidos na entrada e os valores preditos pelo modelo. Minimizar a função de perda é análogo a minimizar o  $\chi^2$  no caso de uma regressão clássica, onde se buscam os parâmetros que melhor descrevem uma determinada distribuição de dados. Como os modelos aprendem com um extenso conjunto de dados, o ajuste dos pesos é feito através da média das funções de perda individuais. O algoritmo de minimização e as funções de perda mais comuns no âmbito de aprendizado de máquina serão detalhados mais a frente, na seção 5.3.

A velocidade em que os pesos são atualizados é definida pela taxa de aprendizado  $\eta$ , responsável por reger a rapidez com que o modelo aprende. No âmbito da otimização do aprendizado, a introdução de um *bias*, ou viés, ajuda o modelo a se ajustar de forma mais precisa aos dados de entrada, além de ser um artifício para lidar com entradas nulas, garantindo a ativação do algoritmo mesmo quando o conjunto de entrada é igual a zero. O *perceptron* foi útil para representar operações lógicas simples. Mas se mostrou ineficiente para resolver problemas não linearmente separáveis. Para lidar com estes problemas, de natureza não linear, surgiram as redes neurais multicamadas (*multilayer perceptron* ou MLP), que possuem uma ou mais camadas ocultas, como mostra o diagrama

da figura 5.3. Inspiradas no funcionamento do cérebro humano, estas estruturas conectam inúmeros neurônios artificiais em múltiplas camadas. Através da implementação de funções de ativação não lineares e o uso de algoritmos de aprendizagem sofisticados, como a retropropagação, as redes neurais se tornaram capazes de processar grandes volumes de dados e aprender padrões complexos. As MLPs inauguraram um novo momento para a inteligência artificial com aplicações diversas, como em problemas de classificação multiclasse e regressão, com a previsão de variáveis contínuas. Estes modelos foram precursores de outros algoritmos de aprendizagem de máquina, como é o caso das redes neurais convolucionais. Elas utilizam filtros convolucionais ao invés de neurônios em suas camadas ocultas, o que as habilita a reconhecer padrões e estruturas locais em imagens.

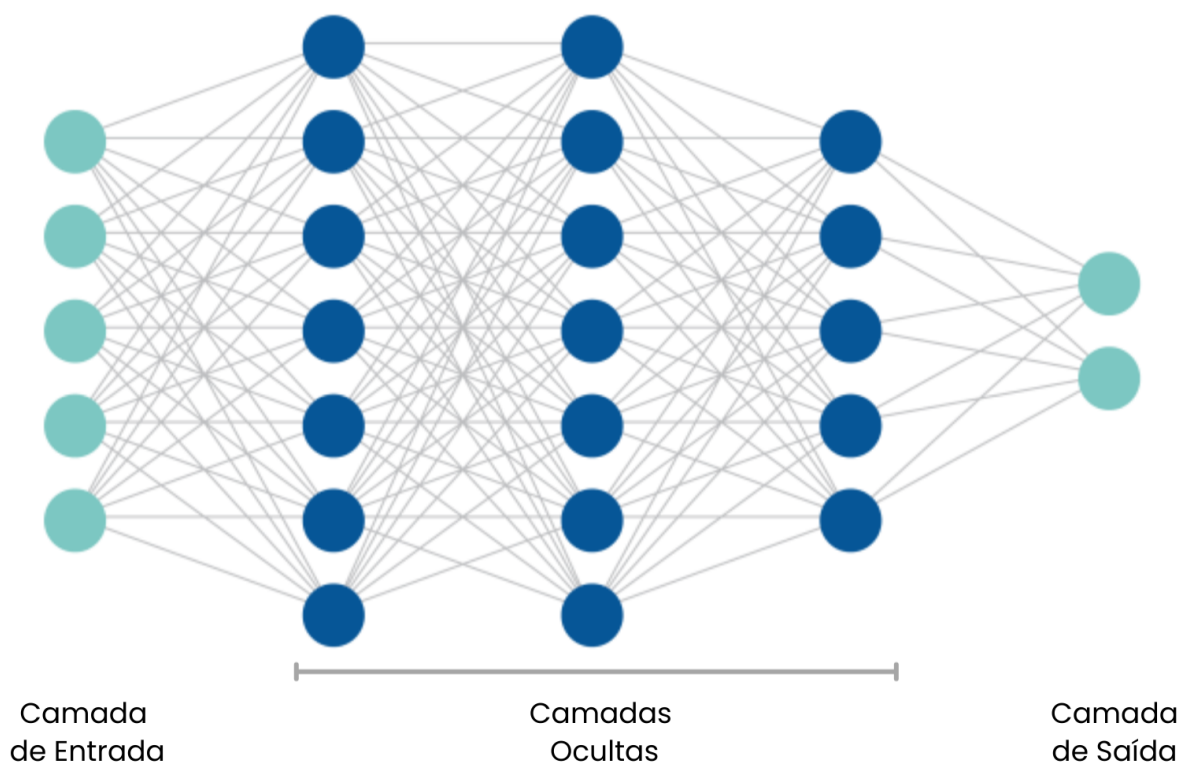


Figura 5.3 – Arquitetura de uma rede neural com três camadas ocultas totalmente conectadas (Figura adaptada de <https://doc.comsol.com>).

## 5.2 REDES NEURAIS CONVOLUCIONAIS

Uma rede neural convolucional (*Convolutional Neural Network* ou CNN) pode receber de entrada objetos com mais de uma dimensão, como imagens, que são representadas por matrizes bidimensionais [61]. Em uma rede totalmente conexa, as entradas são dados independentes, diferentemente de uma imagem, onde as informações de cada pixel estão correlacionadas. A ampla eficiência destas arquiteturas no processamento de imagens é motivada pelo uso de filtros, aplicados em operações convolucionais, para reconhecimento dos padrões. Os filtros levam em conta as correlações dos dados, considerando a posição

dos pixels e dos valores de seus vizinhos para identificar estruturas locais como bordas, intensidades e texturas. As CNNs foram inspiradas no funcionamento do córtex cerebral de mamíferos [62] e são protagonistas em aplicações de visão computacional. Estes modelos são constituídos pela combinação de camadas de convolução, funções de ativação, camadas de *pooling*, *dropout* e camadas totalmente conexas. Nas subseções adiante iremos resumir brevemente o funcionamento de cada um destes blocos e como eles se relacionam.

### 5.2.1 FILTROS CONVOLUCIONAIS

A convolução é uma operação linear que combina duas funções a fim de obter a sobreposição entre elas. Suponha as funções contínuas em uma dimensão  $f(t)$  e  $g(t)$ , a função resultante da convolução destas duas pode ser expressa como:

$$f(t) * g(t) = \int f(\tau)g(t - \tau)d\tau \quad (5.3)$$

onde  $f(t)$  é a entrada,  $g(t)$  é o filtro, também chamado de *kernel* e, no contexto das CNNs o resultado de uma convolução,  $f(t) * g(t)$ , é denominado mapa de características ou *feature map*.

A convolução é uma ferramenta muito utilizada no processamento de imagens, porque através de aplicações de filtros é possível realçar ou atenuar determinadas características dos dados de entrada. Por exemplo, este tipo de operador é capaz de detectar linhas horizontais ou verticais, pontos e bordas nas imagens. No caso de uma imagem (2D) discretizada em pixels, os filtros são representados por matrizes cujos elementos são chamados coeficientes. Eles atuam sequencialmente sobre cada região da imagem redefinindo os valores dos pixels. Para entender esta operação vamos escrever um dado filtro  $G$ , que atuará sobre a região  $F$  de uma imagem.

$$F = \begin{bmatrix} f_1 & f_2 & f_3 \\ f_4 & f_5 & f_6 \\ f_7 & f_8 & f_9 \end{bmatrix} \quad G = \begin{bmatrix} g_1 & g_2 & g_3 \\ g_4 & g_5 & g_6 \\ g_7 & g_8 & g_9 \end{bmatrix} \quad \xrightarrow{F * G} \quad R = \sum_{i=1}^9 f_i g_i$$

O filtro atua sobre regiões que possuem dimensões iguais às suas e para cada região produz um novo pixel, de valor  $R$ , através do somatório dos produtos de elementos da matriz de entrada e dos coeficientes. Ao ser aplicado em todas as regiões da imagem, ou seja, com  $F$  sendo centrado em cada pixel diferente da imagem, esta operação resulta em uma nova matriz, que é o mapa de características. Os pixels próximos à borda requerem condições adicionais que serão discutidas posteriormente. Na figura 5.4 ilustramos o resultado da aplicação de um filtro 2 x 2 em uma imagem (matriz) 3 x 3, resultando em um mapa de características 2 x 2.

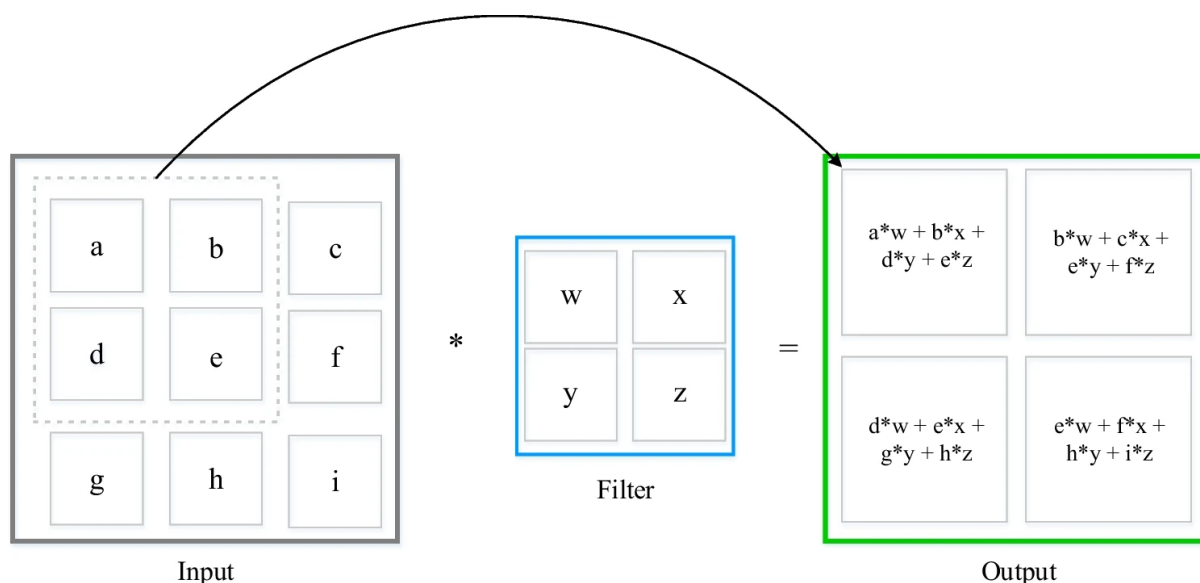


Figura 5.4 – Esquema ilustrativo da aplicação de um filtro de dimensão 2x2 sobre uma matriz (Figura extraída de [63]).

## 5.2.2 CAMADAS CONVOLUCIONAIS

Responsáveis pela aplicação dos filtros convolucionais nos dados de entrada, estas camadas são as grandes protagonistas de uma CNN. Diferentemente de uma camada totalmente conectada, as camadas convolucionais têm conexões esparsas. Isso significa que cada pixel na matriz de entrada só pode influenciar em uma pequena região do mapa de características resultante, já que os filtros atuam em regiões definidas por suas dimensões. Ao contrário, se a camada fosse totalmente conexa, qualquer elemento (pixel) poderia influenciar qualquer saída da camada. Normalmente os filtros convolucionais são bem menores do que as imagens de entrada, o que permite que estes operadores reconheçam padrões mais específicos. Além disso, o número de parâmetros de aprendizagem é reduzido significativamente, pois cada filtro convolucional se conecta localmente aos dados da matriz de entrada, o que não ocorre numa camada densa, onde cada neurônio recebe todos estes dados de uma só vez.

Uma outra propriedade que contribui para a eficiência das camadas convolucionais no processamento de imagens é o compartilhamento dos parâmetros de aprendizagem. Em uma camada densa cada peso é independente e aplicado uma única vez aos dados de entrada. O que não acontece em uma camada convolucional, onde cada elemento de um filtro é aplicado a todos os dados de entrada. Desta forma um único conjunto de parâmetros, os coeficientes do filtro, são utilizados para reconhecer um tipo específico de padrão ao longo de toda a imagem. Esta característica das camadas convolucionais dá à rede invariância translacional, ou seja, o algoritmo é capaz de reconhecer padrões independentemente de sua posição na imagem.

A dimensão do mapa de características gerado é reduzida em N-1 pixels em cada



direção (onde  $N$  é a dimensão do *kernel*) devido ao efeito da borda (ver figura 5.4). Em alguns casos existe o interesse em preservar as dimensões originais da imagem de entrada, incluindo as informações dos pixels das suas bordas na operação de convolução. Existem duas estratégias que podem ser implementadas para controle das dimensões dos mapas de características gerados.

- *Padding*

Consiste na adição de pixels artificiais nos entornos da imagem de entrada. Estes pixels possuem valor nulo e permitem controlar o tamanho do mapa de características, preservando a mesma dimensão da imagem de entrada. Na figura 5.5 temos ilustrados os efeitos da aplicação do *padding* na convolução de um filtro  $3 \times 3$  a uma imagem  $4 \times 4$ .

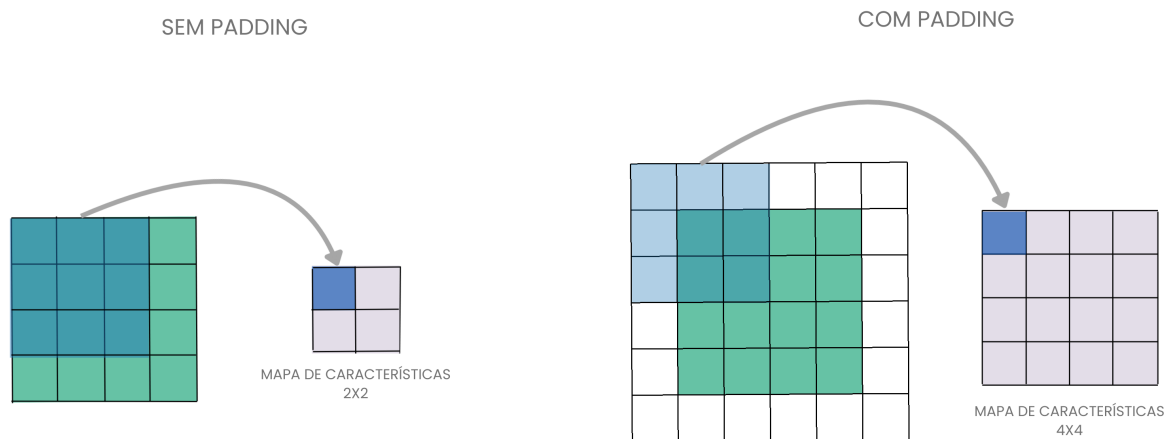


Figura 5.5 – Ilustração dos efeitos do *padding* na convolução para um *kernel*  $3 \times 3$ .

- *Striding*

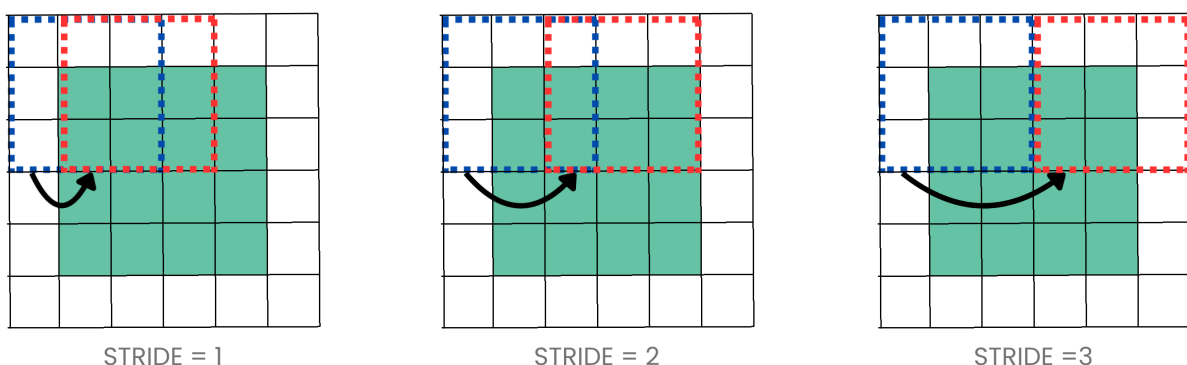


Figura 5.6 – Ilustração dos efeitos do *stride* na convolução.

É o controle do passo que o filtro admite para percorrer a imagem de entrada. Para que o mapa de características produzido na convolução tenha as mesmas dimensões da entrada. O passo mínimo é definido por um *stride* igual a 1, o que significa que o filtro

vai percorrer a imagem se movendo pixel a pixel. Na figura 5.6, temos três exemplos mostrando como o filtro desliza sobre a imagem para *strides* iguais a 1, 2 e 3.

### 5.2.3 CAMADAS DE ATIVAÇÃO

No contexto de uma CNN, as funções de ativação atuam sobre os mapas de características, modulando a informação para que sejam recebidos pelas camadas seguintes. A implementação de funções de ativação não-lineares deu aos algoritmos de aprendizagem de máquina a capacidade de processar e classificar dados cujo comportamento não pode ser modelado por uma função linear. Isto permite que a rede neural seja capaz de aprender padrões complexos e generalizar o reconhecimento destas estruturas para dados não vistos. Existem alguns critérios para definir uma função de ativação tendo em vista a melhora no processo de aprendizagem [64]:

1. Precisa ser uma função com baixo custo computacional para que o processo de aprendizagem do modelo não se torne lento.
2. A função de ativação deve facilitar o fluxo dos parâmetros de aprendizagem, garantindo a convergência do modelo.
3. Não pode distorcer muito a distribuição dos dados de entrada para que a rede reconheça os padrões de forma eficiente.

Existem muitos tipos de funções de ativação, e a depender do problema, se é uma tarefa de classificação ou regressão, a escolha de qual ativação utilizar na última camada da rede é estratégica, como veremos em alguns exemplos a seguir, cujos gráficos estão reportados na figura 5.7.

- *Linear*

Esta função retorna o valor de entrada multiplicado por uma constante  $a$ . Como não introduz não-linearidade, ela normalmente é utilizada na última camada da rede nas tarefas de regressão, já que sua saída é um valor não limitado.

$$\varphi(x) = ax \quad (5.4)$$

- *ReLU*

A *ReLU* ou Unidade Linear Retificada retorna um valor nulo se a entrada for negativa e o próprio valor da entrada caso ela seja positiva. É muito utilizada nas camadas ocultas, pois embora seja parcialmente linear ela ainda é capaz de introduzir não-linearidade à rede e possui baixo custo computacional. A saída desta função são valores pertencentes ao domínio  $[0, \infty[$ .

$$\varphi(x) = \max(0, x) \quad (5.5)$$

- *Sigmóide*

É uma função não-linear muito tradicional que transforma a entrada em um número real dentro do intervalo  $[0,1]$ . É considerada uma função de alto custo computacional por conta da exponencial, por isso é mais comumente utilizada nas camadas de saída em redes voltadas à classificação binária.

$$\varphi(x) = \frac{1}{1 + e^{-x}} \quad (5.6)$$

- *Softmax*

A *softmax* é uma generalização da função *sigmóide* para os casos onde a saída deve ser interpretada como uma probabilidade [65]. Quando temos apenas duas classes, a sua expressão culmina em uma *sigmóide*. Esta função é útil como ativação da última camada para problemas de classificação multiclasse. Já que a soma das saídas para cada classe é sempre 1 por construção, o contradomínio é restrito ao intervalo  $[0,1]$ , sendo 1 o indicativo que a entrada pertence a uma determinada classe  $k$ , e 0 no caso contrário.

$$\varphi(a_k) = \frac{e^{a_k}}{\sum_{k'} e^{a_{k'}}} \quad (5.7)$$

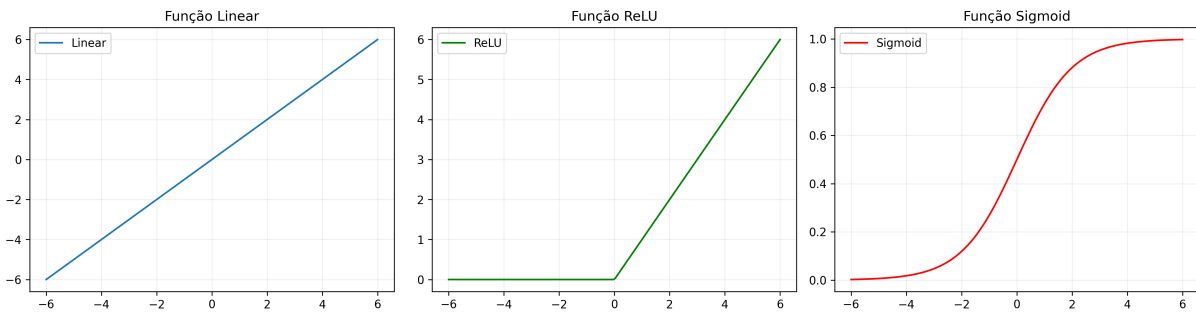


Figura 5.7 – Funções de ativação: Linear, ReLU e *Sigmoid*.

## 5.2.4 CAMADAS DE AGRUPAMENTO

As camadas de agrupamento ou *pooling*, reduzem a dimensionalidade dos mapas de características gerados nas camadas de convolução. As informações extraídas das imagens são resumidas em estruturas de dados menores que facilitam o seu processamento, reduzindo o custo computacional do modelo e simplificam a interpretação dos resultados (significado das saídas). Além disso estas camadas também podem ajudar a introduzir invariância translacional na rede. Existem formas diferentes de condensar os padrões encontrados, as mais comuns são:

- *Max Pooling*

Divide o mapa de características em grupos de elementos de mesmo tamanho. Toma o valor máximo de cada grupo para constituir uma nova estrutura matricial. Na figura 5.8 temos a aplicação desta técnica de agrupamento a uma matriz 4 x 4, resultando em uma saída 2 x 2.

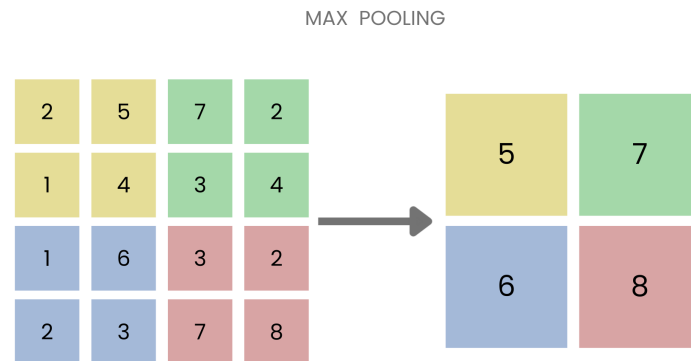


Figura 5.8 – Ilustração dos efeitos do *max pooling* sobre os mapas de características.

- *Average Pooling*

Neste caso, o mapa de características é dividido em grupos e, para criar a nova imagem, a média de cada pequeno grupo de elementos é calculada. Na figura 5.9 temos o resultado deste *pooling* sobre uma matriz 4 x 4.

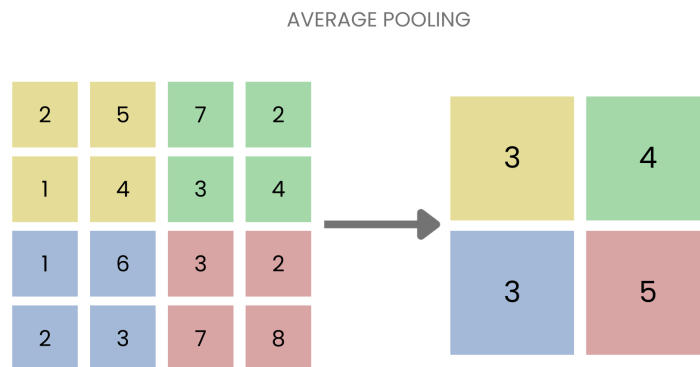


Figura 5.9 – Ilustração dos efeitos do *average pooling* sobre os mapas de características.

A arquitetura típica de uma rede neural é mostrada na figura 5.10. A parte responsável pela extração de padrões e características é construída com camadas convolucionais, funções de ativação e camadas de agrupamento. No segmento de decisão, são associadas camadas de *flatten*, que irão unidimensionalizar os dados recebidos pela camada anterior, camadas totalmente conexas e uma função de ativação na camada de saída, que pode mudar a depender do tipo do problema (classificação ou regressão).

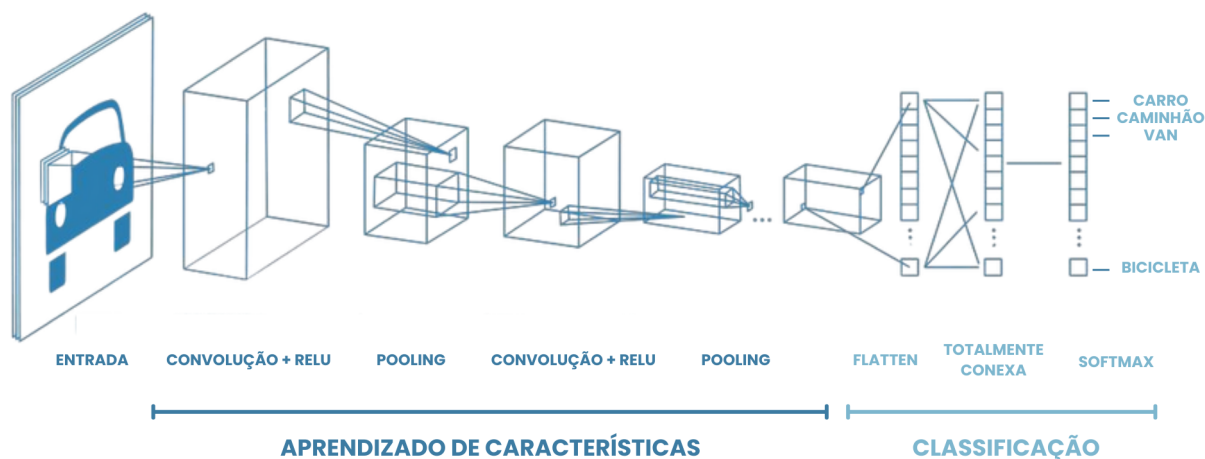


Figura 5.10 – Arquitetura típica de uma rede neural convolucional que combina todos os tipos de camadas anteriormente mencionados: camadas de convolução, ativação e agrupamento. Figura adaptada de (<https://it.mathworks.com/discovery/convolutional-neural-network.html>).

## 5.3 APRENDIZAGEM DA REDE NEURAL

Existem basicamente dois tipos de aprendizagem de máquina: supervisionado e não supervisionado. No caso supervisionado a rede recebe um conjunto de imagens e um rótulo relacionado a cada uma. A proposta é que a rede possa reconhecer os padrões em cada imagem e associar aquele conjunto de características a um determinado rótulo. Aqui, o processo de aprendizado é supervisionado pois a rede neural possui um gabarito para comparar com sua saída, por conta disso é um método muito utilizado em problemas de classificação ou regressão. Há também o aprendizado não supervisionado, onde a rede recebe apenas o conjunto de dados e precisa identificar os padrões sem interferência humana, criando uma relação entre padrões parecidos. Este tipo de aprendizado é útil em tarefas de *clusterização*, onde a rede é utilizada para agrupar dados com características parecidas.

A metodologia implementada neste trabalho é um aprendizado supervisionado, por isso no decorrer das próximas sessões vamos nos dedicar a entender o processo de aprendizagem de uma rede neural convolucional neste modo de operação.

### 5.3.1 FUNÇÕES DE PERDA

As funções de perda, também conhecidas como *loss functions* são fundamentais no processo de aprendizagem supervisionado. É através delas que a rede compara os resultados esperados, provenientes dos rótulos, com os resultados de saída. As funções de perda representam o quanto as previsões ou classificações do modelo estão distantes do valor esperado. Por isso, a rede vai buscar o melhor conjunto de parâmetros de forma a minimizar esta função, tornando seus resultados mais acurados. As funções que permitem que o

modelo convirja para uma solução e a escolha delas varia a depender do tipo de tarefa a ser implementada. As funções de perda mais comuns para problemas de regressão, onde a rede deve prever um valor, são a MAE (*Mean Absolute Error*) e MSE (*Mean Square Error*) [66].

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \bar{y}_i| \quad (5.8)$$

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \bar{y}_i)^2 \quad (5.9)$$

onde

$N$  é o número de exemplos disponíveis para treinamento

$y_i$  é o valor do rótulo

$\bar{y}_i$  é a saída prevista pelo modelo

Em tarefas de classificação, a função de perda mais comum é a função de entropia cruzada ou *Cross Entropy Loss*. A entropia cruzada fornece a probabilidade de uma entrada pertencer a determinada classe, e por isso sua saída é restrita ao intervalo  $[0,1]$ . Quanto mais próximo o valor da saída é de 1 ou de 0, maiores as chances de pertencer a uma das duas classes.

$$CE = - \sum_{i=1}^N y_i \log \bar{y}_i \quad (5.10)$$

### 5.3.2 RETROPROPAGAÇÃO

O aprendizado de uma rede neural se dá através da minimização da função de perda. Após comparar os resultados previstos e esperados, o modelo ajusta os parâmetros da rede, i.e. os coeficientes dos filtros convolucionais e dos neurônios das camadas totalmente conectadas, de modo a diminuir a perda. Como será descrito nas seções subsequentes, esse processo é realizado iterativamente até atingir uma convergência, obtendo os parâmetros que minimizam a perda e, portanto, fornecem as melhores previsões para o modelo.

No entanto, há uma dificuldade prática para realizar essa minimização devido à presença de muitas camadas e à forma das funções de ativação que eram utilizadas nas primeiras redes neurais. A minimização requer o cálculo de derivadas. No caso das redes neurais, esse cálculo pode ser lento e levar à amplificação dos erros numéricos no cálculo das derivadas, que se acumulam em cada camada, já que a função que mapeia as entradas às saídas é muito complexa (concatenação dos neurônios). Por um lado, existe o problema dos gradientes nulos, ou muito pequenos. Isso ocorre porque os gradientes são multiplicados várias vezes por derivadas de funções de ativação (como sigmoid ou tanh), que possuem derivadas quase nulas na maior parte do domínio. Também pode acontecer o inverso: os gradientes crescem exponencialmente. Isso pode ocorrer quando os pesos têm valores grandes ou quando as

derivadas das ativações são grandes. Esses dois problemas essencialmente impediam o aprendizado da rede na prática, além de torná-lo muito lento. Se os gradientes se anulam, a rede para de aprender e se eles explodem os pesos se tornam enormes em poucas iterações, a rede diverge ou fica instável. Associado a essas questões, existia o chamado “problema do crédito”: como saber quanto cada peso em cada camada contribuiu para o erro final? Sem isso não é possível ajustar adequadamente os pesos e aprender.

A solução para esse problema veio a través de uma ideia muito simples e poderosa: utilizar a regra da cadeia do cálculo diferencial para calcular os gradientes da função de perda em relação a cada peso. Ou seja, as derivadas são calculadas camada por camada e reutilizam os gradientes locais já calculados para as camadas anteriores. Isso torna o cálculo dos gradientes eficiente: o custo computacional cresce de forma linear com o número de pesos, em vez de exponencial com o número de camadas. Além disso, ajuda a diminuir a propagação dos erros nos gradientes que levava ao problema dos gradientes que se anulam ou divergem. O cálculo das derivadas é realizado de trás para a frente, ou seja, propagando as derivadas da saída da rede em direção à entrada, reutilizando cálculos intermediários. Daí o nome de retropropagação.

Para ilustrar como o método funciona vamos supor uma rede simplificada com três unidades de processamento cujas duas últimas saídas de ativação são dadas por  $a^{(L-1)}$  e  $a^{(L)}$ . O resultado da última saída pode ser expresso por:

$$a^{(L)} = f(w^L a^{(L-1)} + b^L) \quad (5.11)$$

onde

$f$  é a função de ativação  
 $w^L$  é o filtro da última saída  
 $b^L$  é o viés aplicado na última saída

A função de perda pode ser definida por:

$$L_0 = (a^{(L)} - \bar{y})^2 \quad (5.12)$$

As derivadas parciais da função de perda são calculadas com respeito a cada parâmetro. Como a análise é retropropagada, obtemos a forma como a saída da camada anterior influencia nesta função:

$$\frac{\partial L_0}{\partial a_k^{(L-1)}} = \sum_{j=0}^{n_L-1} \frac{\partial z_j^L}{\partial a_k^{(L-1)}} \frac{\partial a_j^{(L)}}{\partial z_j^{(L)}} \frac{\partial L_0}{\partial a_j^{(L)}} \quad (5.13)$$

O desenvolvimento do método de retropropagação ocorreu na década de 1980 [67] e foi crucial para tornar prática a utilização de redes neurais multi-camadas, que são os embriões do *deep learning* atual. Outros aspectos que tornaram mais eficiente e robusto o treinamento das redes foram: a utilização de funções de ativação que preservem melhor

os gradientes, como ReLU, em vez de sigmoide e tanh puros; a inicialização adequada de pesos e o uso de otimizadores adaptativos, como ADAM e RMSProp, que veremos a seguir.

### 5.3.3 ALGORITMOS DE MINIMIZAÇÃO DA FUNÇÃO DE PERDA

Existem vários algoritmos para realizar a minimização da função de perda utilizando a retropropagação. Um dos mais simples e mais utilizados é o método do gradiente descendente, cuja essência está em atualizar os parâmetros da rede na direção oposta à do crescimento da função de perda. Os filtros aplicados na primeira iteração de uma CNN são definidos de forma aleatória e depois são atualizados de acordo com o gradiente da função de perda. Sabe-se que o gradiente de uma função aponta na direção de seu crescimento. Como o intuito do algoritmo é minimizar o erro, a atualização dos filtros é feita na direção negativa do gradiente. A expressão matemática para atualização de um parâmetro aprendível é dada por:

$$w_{t+1} = w_t - \eta \nabla f(w_t) \quad (5.14)$$

onde

$\eta$  é a taxa de aprendizagem, que define o passo da minimização

$w_{t+1}$  é o valor do parâmetro na iteração seguinte

$w_t$  é o parâmetro na iteração atual

$f(w_t)$  é a função de perda

Quando esta atualização de pesos é aplicada a todo conjunto de dados de uma só vez, ela recebe o nome de *Batch Gradient Descent*. Mas ela também pode ser aplicada a pequenos lotes de dados por vez e nestes casos é chamada *Mini Batch Gradient Descent*. Este último modo de atualização é o mais utilizado no treinamento de redes neurais, já que para grandes conjuntos de dados, a memória necessária para realizar o processo de minimização pode facilmente exceder o disponível.

Para melhorar a convergência dos modelos, foram desenvolvidas estruturas que buscam aumentar a eficiência na minimização da função de perda. Dentre elas podemos citar o uso de taxas de aprendizagem, *momentum* e dos métodos adaptativos [68] [69].

- *Momentum*

É um método de atualização de pesos inspirado no conceito de inércia da física. Esta estrutura utiliza parte das atualizações anteriores combinando-as com o novo peso através de um fator multiplicativo,  $\rho$ , que chamamos de *momentum*. Este mecanismo atua acelerando o aprendizado ao longo de direções que o gradiente aponta de forma consistente e suaviza nos casos onde ele oscila.



- *Nesterov Momentum*

Neste método, o gradiente é calculado num passo à frente, na posição deslocada pelo termo de *momentum*. Nós prevemos onde este termo vai nos levar e corrigimos o passo de atualização antecipadamente, o que resulta em um ajuste de pesos mais preciso.

- *RMSprop*

Este algoritmo utiliza um método adaptativo de ajuste de pesos, com ele as taxas de aprendizado são atualizadas automaticamente, se adaptando ao gradiente da função de perda durante o treinamento. O *RMSprop* faz o ajuste da taxa de aprendizagem de acordo com o histórico do gradiente para cada componente do peso. Desta forma aqueles parâmetros com histórico reduzido são atualizados com uma taxa de aprendizagem maior, enquanto que os parâmetros com histórico maior são atualizados mais lentamente, com uma taxa de aprendizado menor. O diferencial deste método está no armazenamento dos gradientes anteriores. Diferentemente de outros métodos adaptativos, o *RMSprop* guarda os gradientes como uma média exponencialmente decrescente dos quadrados de cada gradiente.

- *Adam*

*ADAM (Adaptive Moment Estimation)* é um outro método baseado em taxa de aprendizagem adaptativa que, assim como *RMSprop*, guarda os gradientes passados como uma média que decresce exponencialmente. Mas, diferentemente dos outros algoritmos, *ADAM* usa estimativas de primeiro e segundo momento dos gradientes da função de perda para atualização dos pesos.

- *Nadam (Nesterov-accelerated Adaptive Moment Estimation)*

Este otimizador é a combinação do *ADAM* com o *Nesterov Momentum*, proposto visando convergências mais rápidas e a melhora da performance dos modelos [70].

## 5.4 TREINAMENTO, VALIDAÇÃO E TESTE

A primeira etapa para treinar adequadamente uma rede neural consiste na divisão do *dataset* em três conjuntos: treinamento, validação e teste. Cada um destes conjuntos tem uma finalidade e contribui de uma forma específica para compreender e melhorar a performance da rede neural.

O conjunto de treinamento será utilizado para o ajuste dos pesos ou filtros das camadas ocultas. É através dele que a rede neural vai aprender os padrões nos dados para fazer as previsões e classificações. Por este motivo, ele deve representar a maior parte do *dataset*,

conforme representado pelas proporções na figura 5.11. Depois de aprender com os dados, a performance da rede é avaliada através do conjunto de validação. Com ele podemos fazer mudanças na arquitetura do modelo para otimizar os resultados. As estruturas que podem ser alteradas são chamadas de hiperparâmetros. É no processo de validação que identificamos problemas de aprendizagem como o *overfitting*, que ocorre quando o modelo se superajusta aos dados e acaba aprendendo ruídos e exceções. Nestes casos, a rede perde a capacidade de generalização quando apresentada a exemplos novos. Já o *underfitting* acontece quando o modelo não é complexo o suficiente para identificar os padrões e não se ajusta devidamente aos dados.

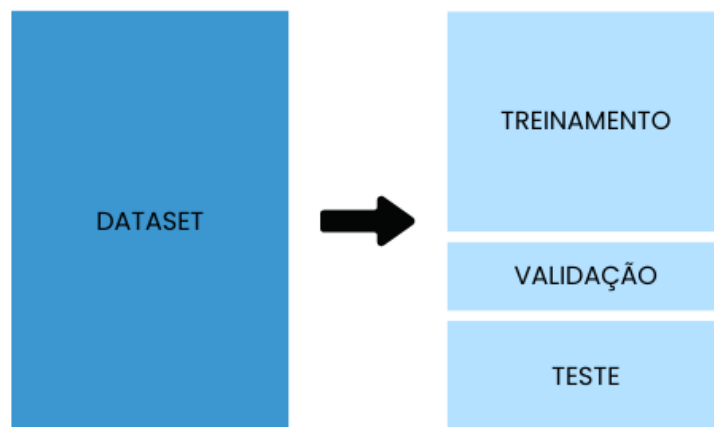


Figura 5.11 – Representação de como é dividido o *dataset*: o conjunto de treinamento deve ser o maior dentre os três, seguido do conjunto de teste.

O treinamento ocorre em iterações chamadas de épocas (*epochs*). O número de épocas determina a quantidade de vezes que todo o conjunto de treinamento será processado pela rede neural. Este processamento é feito dividindo os dados em pequenos subconjuntos, que são os lotes ou *batches*. Para exemplificar podemos considerar um conjunto de treinamento com 5 mil imagens, se cada *batch* possui 25 imagens então teremos 200 *batches*. Se o número de *epochs* é igual a 20, isto quer dizer que o modelo verá o conjunto completo 20 vezes, e em cada vez vai processar os 200 *batches*. O número de *epochs* é um hiperparâmetro relacionado à convergência do modelo, já que com muitas *epochs* o modelo tem maiores chances de aprender de forma eficiente com os dados. Enquanto que, o número e tamanho dos *batches* está atrelado à memória disponível para processar muitos dados simultaneamente. Quanto maior a memória, maior pode ser o tamanho do *batch*. Por outro lado, quanto menor o tamanho do *batch*, mais *batches* serão necessários. Depois de treinado e validado, o passo seguinte é avaliar a capacidade de generalização do modelo para dados nunca antes vistos, isto é feito através do conjunto de teste.

### 5.4.1 CROSS VALIDATION

Um metodologia para avaliar como a performance de um dado modelo depende do conjunto de dados utilizado no treinamento é a validação cruzada (*Cross Validation*). Nesta técnica o *dataset* é dividido em subconjuntos de mesmo tamanho, chamados de *folds*. A cada rodada um dos *folds* é utilizado como conjunto de validação enquanto os demais são destinados ao treinamento. A essência da validação cruzada está em mudar os *folds* destinados ao treinamento em cada iteração, desta forma a capacidade de generalização do modelo é testada e entendida através da performance média obtida. Este processo está esquematizado na figura 5.12.

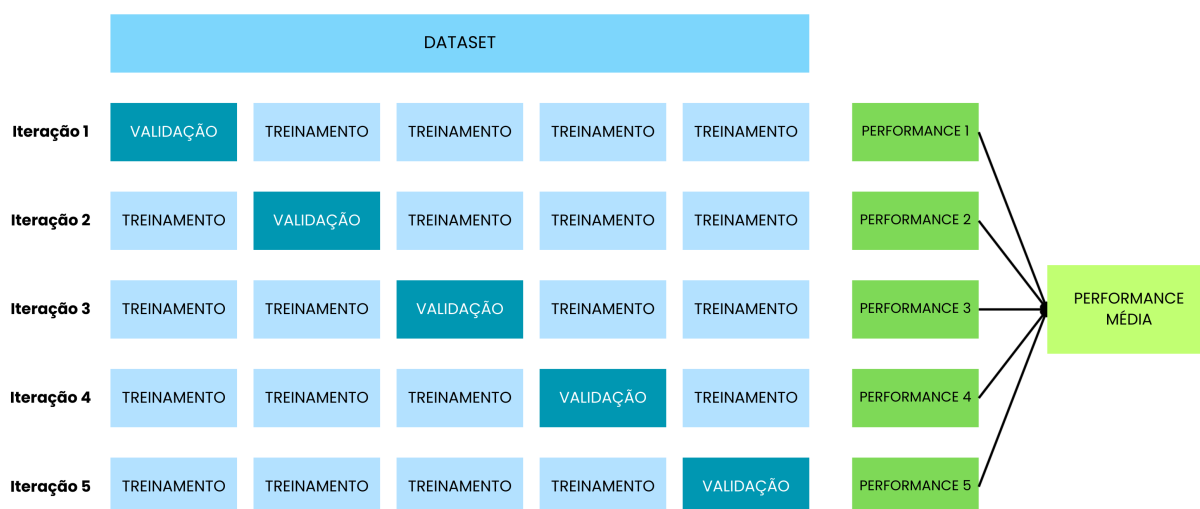


Figura 5.12 – Esquematização da técnica de validação cruzada: a cada iteração um novo *fold* é utilizado como conjunto de teste.

### 5.4.2 TÉCNICAS DE REGULARIZAÇÃO

Alguma técnicas podem ser implementadas com a finalidade de otimizar a performance da rede neural e resolver problemas como o *overfitting*.

- *Dropout*

Uma determinada porcentagem de filtros é desativada aleatoriamente ao longo do treinamento a cada *batch*. Com isso, a rede se baseia em um espectro bem distribuído de informações para tomar uma decisão e não fica dependente de estruturas ou padrões muito específicos.

- *Early Stopping*

Consiste em monitorar o desempenho da rede durante a validação, o que é feito através da função de perda, e estabelecer uma tolerância de *epochs* para as quais não há melhora na performance do modelo. Se a função de perda não diminuir ao longo de

N *epochs* consecutivas entende-se que o procedimento de minimização (aprendizagem) convergiu e o treinamento é interrompido. Normalmente, um critério de tolerância razoável implementada é de 10 *epochs*.



# Capítulo 6

## Resultados

O algoritmo de simulação desenvolvido e apresentado no capítulo 4 foi empregado no treinamento e validação de redes neurais convolucionais utilizadas para prever os valores de profundidade dos eventos de carga pontual a partir de seu padrão de difusão. A fim de encontrar a melhor arquitetura, promovemos um estudo extensivo de diversos modelos variando hiperparâmetros, tais como número de camadas, filtros e neurônios; taxas de *dropout* e otimizadores. Este estudo será detalhado nas seções seguintes, onde também reportamos a comparação dos desempenhos da rede escolhida e do algoritmo utilizado atualmente na análise de dados do CONNIE.

### 6.1 *DATASET*

Para treinar e validar a rede neural criamos um *dataset* com 30 mil imagens contendo um único evento de carga pontual. Estas imagens foram geradas através do algoritmo de simulação desenvolvido e detalhado no capítulo 4. Para compor o plano de fundo, definimos o ruído de leitura como  $0.2 e^-$  e a taxa de emissão de eventos de elétron único de  $0.1 e^-/px/dia$ . Esta combinação de parâmetros nos ajuda a emular o cenário mais realístico dos sensores utilizados no experimento CONNIE. O tamanho das imagens foi definido a partir de um estudo feito com 5 mil imagens de 20 x 20 pixels, com um evento de carga pontual centrado na posição (10,10) e simulado com máxima difusão (evento gerado na profundidade máxima dos sensores, isto é  $675 \mu m$ ) e energia (1015 eV). O intuito do estudo foi verificar os valores das coordenadas mínimas e máximas para os padrões de difusão destes eventos. Depois de simuladas, estas imagens foram submetidas ao algoritmo de extração do CONNIE que nos permitiu catalogar evento por evento num arquivo do tipo *ROOT*. A distribuição dos valores máximo e mínimo para os pixels nos eixos X e Y é reportada na figura 6.1.

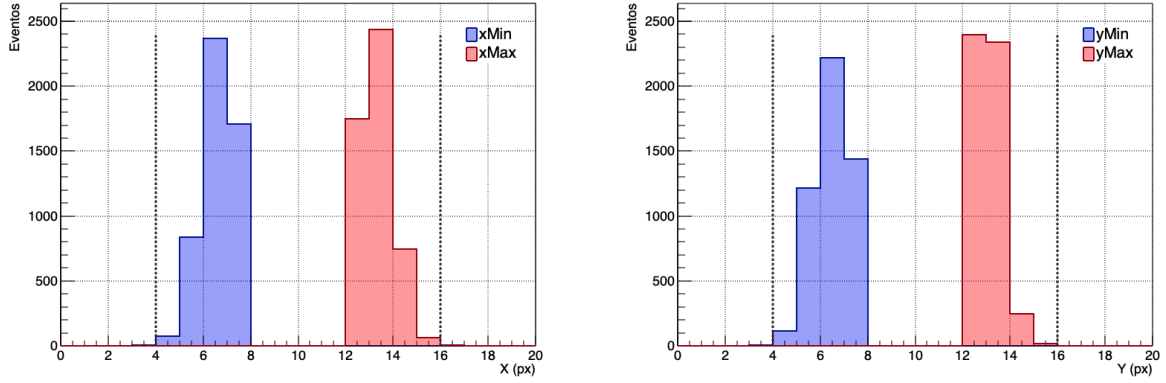


Figura 6.1 – Gráficos das distribuições de valores mínimo e máximo dos pixels nos eixos X (esquerda) e Y (direita). As linhas tracejadas representam a delimitação de uma imagem de 12 x 12 pixels.

Com base nestes resultados, a dimensão das imagens foi definida como 12 x 12 pixels. Cada uma destas imagens foi salva em um *dataset* associado com as respectivas profundidades simuladas.

## 6.2 PRÉ PROCESSAMENTO

O *dataset* utilizado neste trabalho, com 30 mil imagens, foi dividido da seguinte forma: 16 mil foram destinadas para treinamento, 4 mil para validação e 10 mil para teste. As redes neurais têm melhores desempenhos quando recebem os dados de entrada normalizados. Uma técnica muito comum para isso é a chamada normalização *Z-score* ou padronização, que utiliza a média e o desvio padrão dos dados.

$$Z = \frac{X - \mu}{\sigma} \quad (6.1)$$

onde

$X$  é o dado a ser normalizado

$\mu$  é a média da distribuição do conjunto

$\sigma$  é o desvio padrão da distribuição

Para evitar o vazamento de dados (*data leakage*), a média e o desvio padrão devem ser tomados a partir do conjunto de treinamento. Isto garante que as predições feitas no conjunto de teste serão não enviesadas e confiáveis.

## 6.3 BUSCA DE UMA CNN ADEQUADA PARA O PROBLEMA

Para determinar a rede neural convolucional com melhor desempenho nas predições das profundidades dos eventos de carga pontual, um amplo estudo de arquiteturas e

combinações de hiperparâmetros foi feito. Esta investigação pode ser compreendida em sete diferentes rodadas onde a influência de determinados hiperparâmetros no desempenho da rede foi explorada. A cada rodada um grupo de modelos é comparado e o que tem melhor performance segue para a rodada seguinte onde um novo hiperparâmetro é avaliado.

Um conjunto de hiperparâmetros comuns foi utilizado em todos os modelos até a rodada E, eles estão descritos na tabela 1.

Hiperparâmetro	Valor
<i>Batch Size</i>	32
<i>Epochs</i>	100
Optimizador	ADAM
<i>Learning Rate (default)</i>	1e-3
<i>Early Stopping</i> (tolerância)	10
Função de Perda	MSE
Métrica	MAE

Tabela 1 – Hiperparâmetros utilizados no treinamento das redes neurais até a rodada E.

### 6.3.1 MODELOS A

Nesta primeira rodada a arquitetura dos modelos foi estudada quanto ao número de camadas convolucionais e *poolings* a fim de determinar a melhor maneira de combinar estas estruturas. Nos diferentes modelos, preservamos em 20 o número de filtros das camadas convolucionais.

#### MODELO A.1

Nesta arquitetura, representada pelo diagrama 6.2, foram associadas 2 camadas convolucionais intercaladas à outras 2 camadas de *maxpooling*.

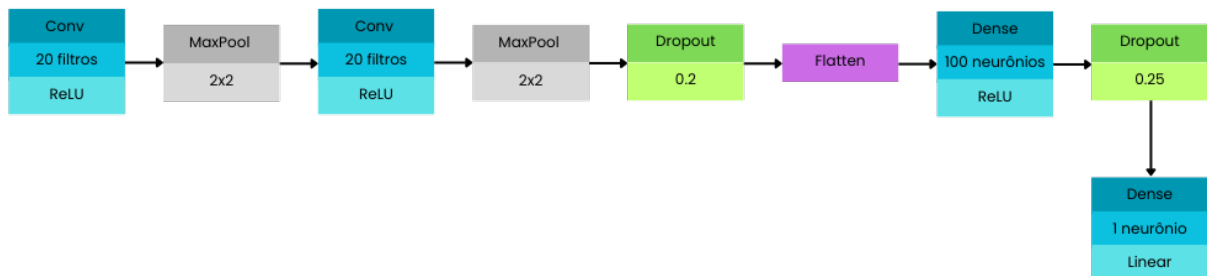


Figura 6.2 – Arquitetura do Modelo A.1

A figura 6.3 reporta as curvas da função de perda (à esquerda) e métrica (à direita) ao longo das *epochs* durante o treinamento e validação deste modelo.



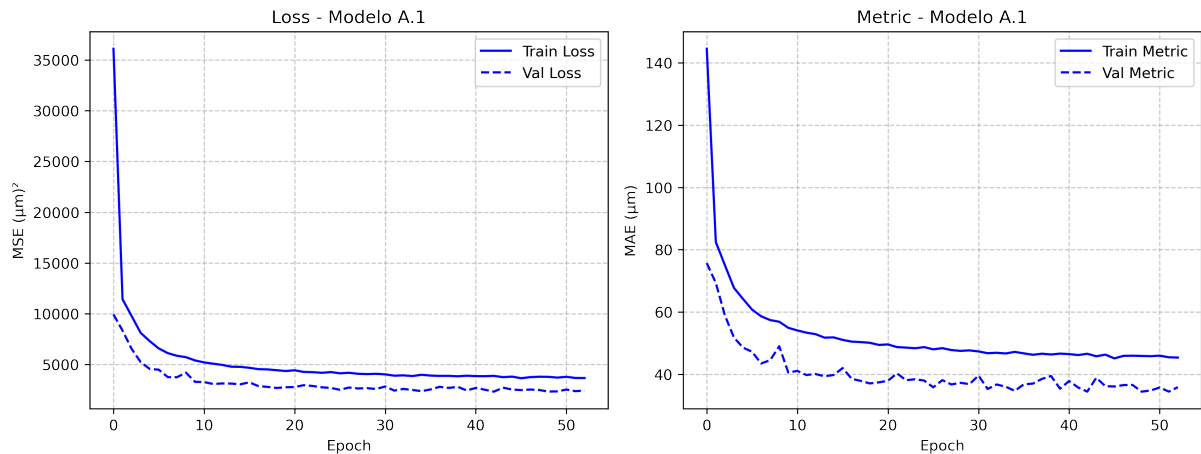


Figura 6.3 – Curvas de desempenho ao longo do treinamento e validação para o Modelo A.1.

A evolução da função de perda e da métrica ao longo das *epochs* nos casos de treinamento e validação revela um pequeno *underfitting*, manifesto em melhores valores das curvas de validação nos dois casos. O modelo converge rápido, utilizando cerca de 50% das *epochs* estipuladas em decorrência do uso do *early stopping*.

## MODELO A.2

Este modelo combina 4 camadas convolucionais a 2 camadas de *maxpooling*, como mostra o diagrama 6.4.

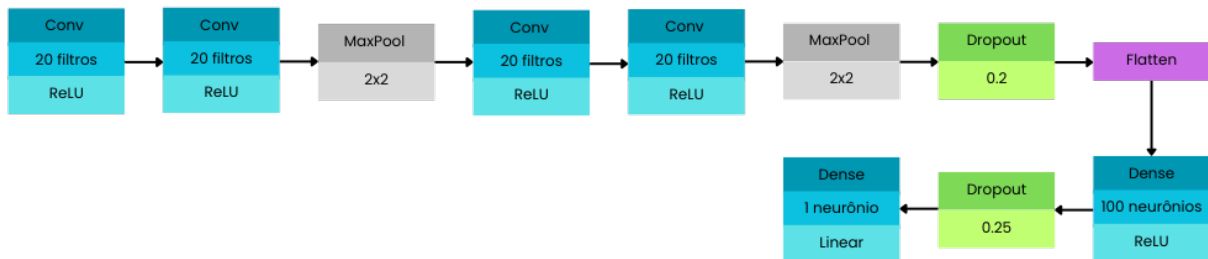


Figura 6.4 – Arquitetura do Modelo A.2.

As curvas da função de perda e da métrica ao longo das *epochs* para a validação e treinamento são apresentadas na figura 6.5. Estes gráficos mostram que também temos melhores valores, para ambas as funções, durante a validação. Este modelo também converge rápido, com aproximadamente 34% das *epochs* disponíveis.

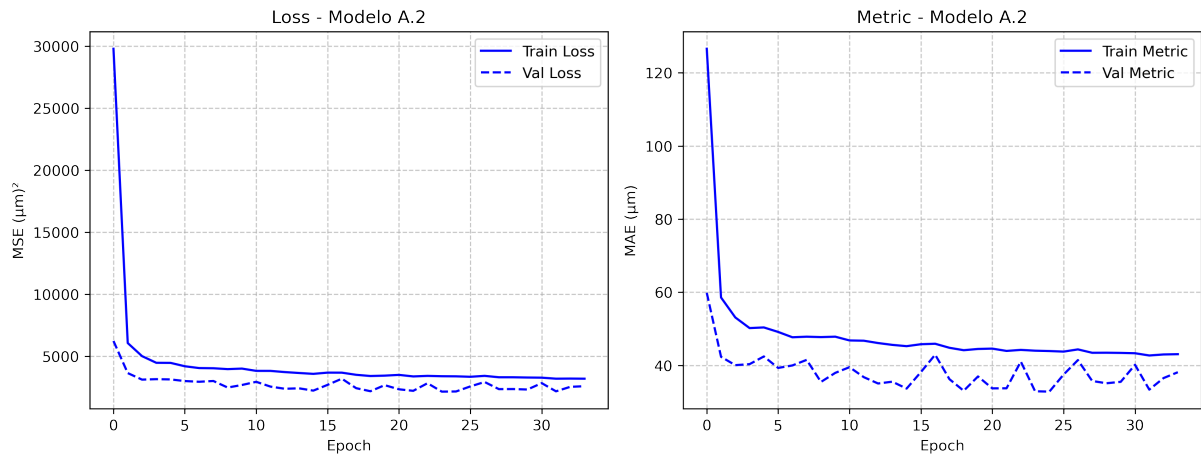


Figura 6.5 – Curvas de desempenho ao longo do treinamento e validação para o Modelo A.2.

### MODELO A.3

Neste modelo 4 camadas convolucionais foram combinadas a 1 camada de *maxpooling*. Estas estruturas foram associadas como o mostra o diagrama 6.6.

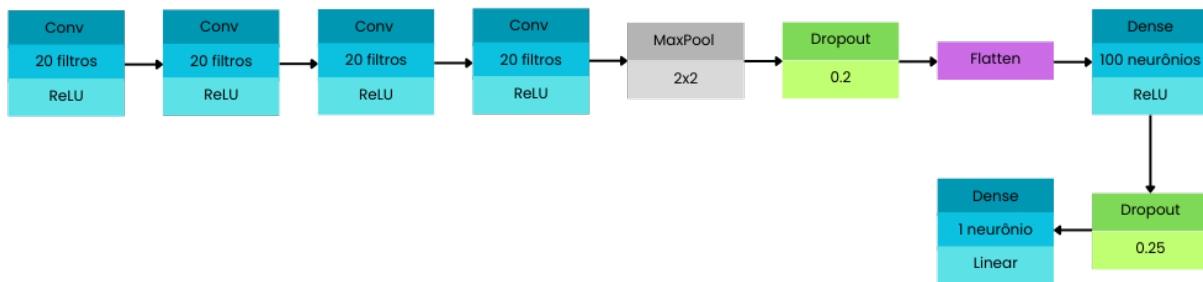


Figura 6.6 – Arquitetura do Modelo A.3.

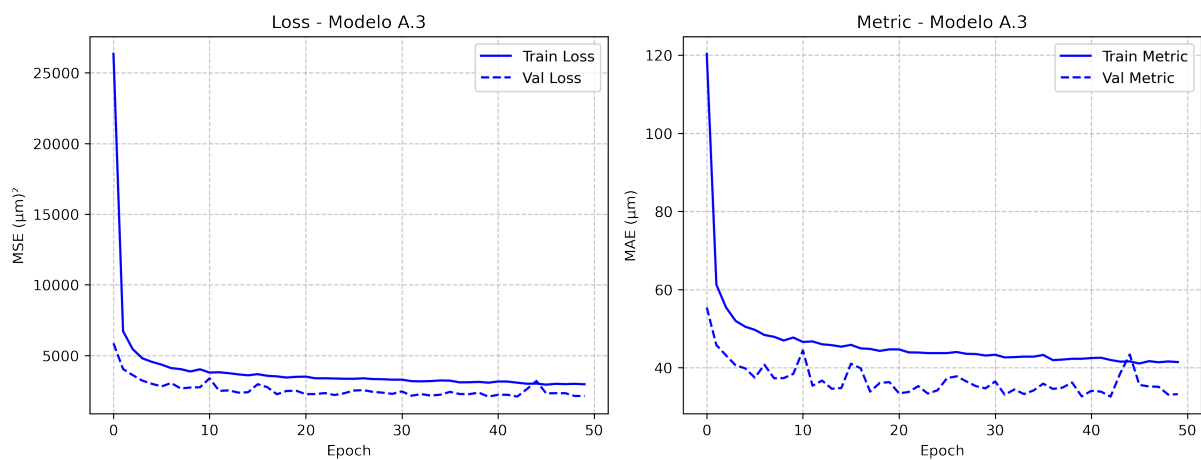


Figura 6.7 – Curvas de desempenho ao longo do treinamento e validação para o Modelo A.3.

A evolução das funções de perda e métrica, reportadas na figura 6.7, também revelam um pequeno *underfitting*, já que as curvas de validação tem melhores valores nos dois casos. O modelo converge com pouco menos de 50% das *epochs*, ou seja, ele demora mais que o anterior para concluir o treinamento/validação.

#### MODELO A.4

Neste modelo 6 camadas convolucionais são intercaladas com 2 camadas de *maxpooling*, como apresentado pelo diagrama 6.8.

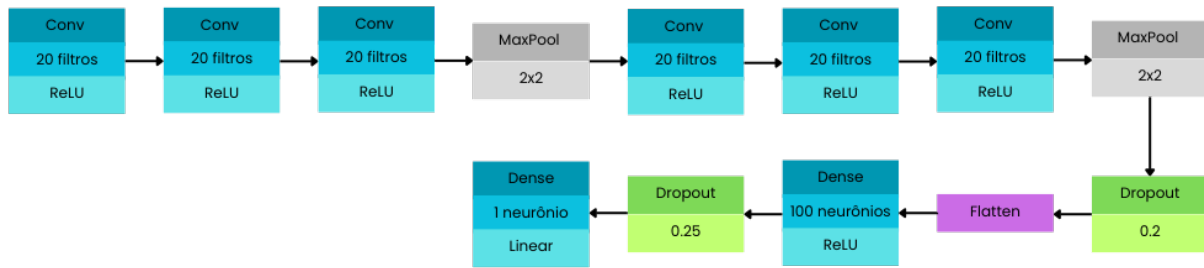


Figura 6.8 – Arquitetura do Modelo A.4.

As curvas da função de perda e da métrica ao longo das *epochs* durante o treino e a validação estão reportadas na figura 6.9.

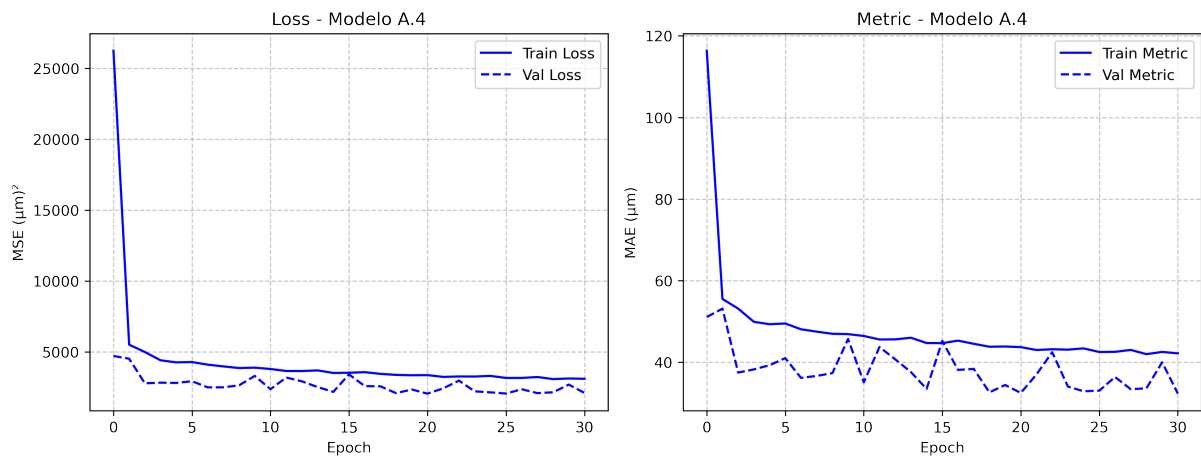


Figura 6.9 – Curvas de desempenho ao longo do treinamento e validação para o Modelo A.4.

Os gráficos mostram um *underfitting* durante o treinamento e validação, pois as curvas de validação são levemente melhores do que as de treinamento. O modelo utiliza 30% das *epochs* para convergir.

#### COMPARAÇÃO DOS MODELOS

Os valores médios da função de perda e da métrica durante a validação para cada um dos modelos está reportado na tabela 2.

Modelo	MSE ( $\mu m$ ) <sup>2</sup>	MAE ( $\mu m$ )
A.1	2316	34.47
A.2	2161	32.92
A.3	2083	32.68
A.4	2047	32.49

Tabela 2 – Valores da função de perda e métrica da validação para os modelos da rodada A.

Para comparar os modelos de cada rodada vamos utilizar o espectro de profundidades reconstruídas, utilizando o conjunto de testes, e agrupar os valores em *bins* de largura igual a  $20 \mu m$ . Com isso teremos 35 *bins*, que percorrem o intervalo de 0 a  $680 \mu m$ . A média e o desvio padrão de cada *bin* são tomados para ajudar na visualização da distribuição dos dados. O *Profile Plot* é feito com a relação das médias de valores preditos pelos modelos, por *bins*, em função dos valores simulados. O desvio padrão dos resultados em cada *bin* é representado através de barras de erro. Para construir o *Residual Plot*, fazemos a diferença entre as médias dos valores preditos e valores simulados. E o desvio padrão representa a largura das barras de erro para cada ponto do *Profile Plot* em função dos valores simulados.

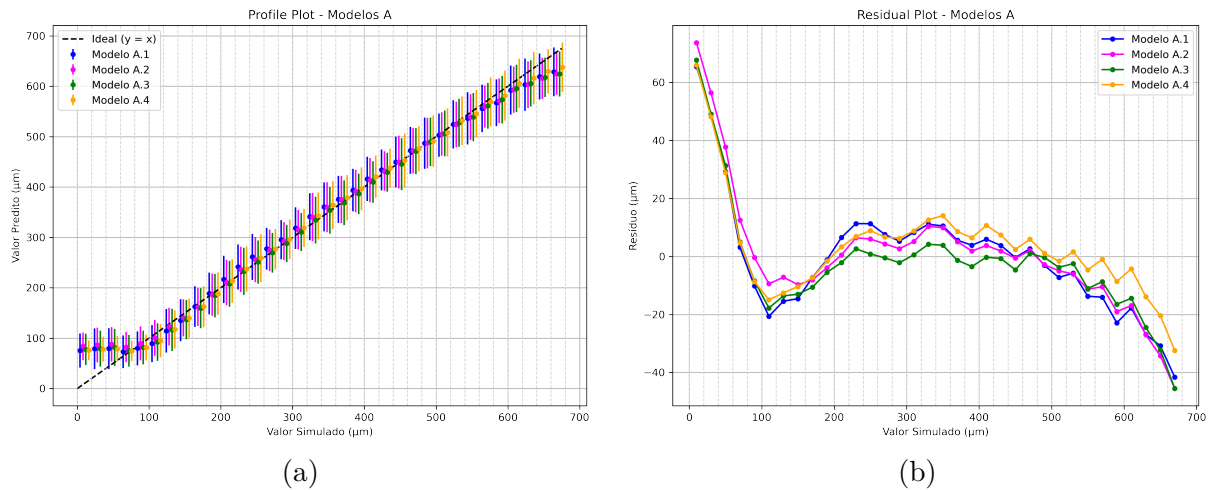


Figura 6.10 – *Profile Plot* (a) e *Residual Plot* (b) para os Modelos A.

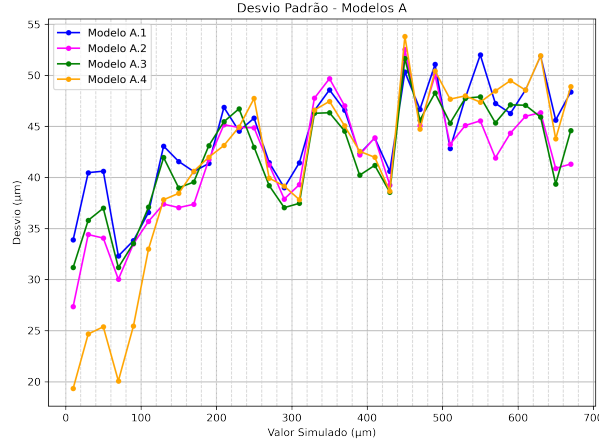


Figura 6.11 – Desvio padrão por *bin* dos Modelos A.

Os gráficos de *profile* e *residual*, reportados na figura 6.10, nos fornecem uma análise conjunta do quão precisas e acuradas são as previsões feitas por cada modelo. O *Residual Plot* evidencia que o modelo A.3 é o que possui melhor desempenho nas regiões de 200 a 500  $\mu\text{m}$ , mas logo se distancia da reta esperada, da mesma forma que A.1 e A.2. Embora não tenha a melhor performance em profundidades intermediárias, o modelo A.4 é o que prevê valores maiores que 500  $\mu\text{m}$  com maior acurácia. O gráfico de desvio padrão, reportado na figura 6.11, mostra que, para eventos rasos, as previsões do modelo A.4 possuem menos dispersão em comparação com os demais. Depois as curvas crescem até atingir um desvio máximo de cerca de pouco menos que 50  $\mu\text{m}$ . Analisando o comportamento destes gráficos comparativos e do desempenho individual de cada modelo, a arquitetura A.4 foi escolhida para a próxima rodada. Este modelo também é o que possui menores valores de MSE e MAE.

### 6.3.2 MODELOS B

Nesta rodada, o hiperparâmetro estudado foi o número de filtros das camadas convolucionais. A estrutura da rede A.4 foi mantida quanto aos demais parâmetros. No diagrama da 6.12 temos a arquitetura comum aos modelos do tipo B, em amarelo estão destacados os hiperparâmetros que serão variados nas arquiteturas.

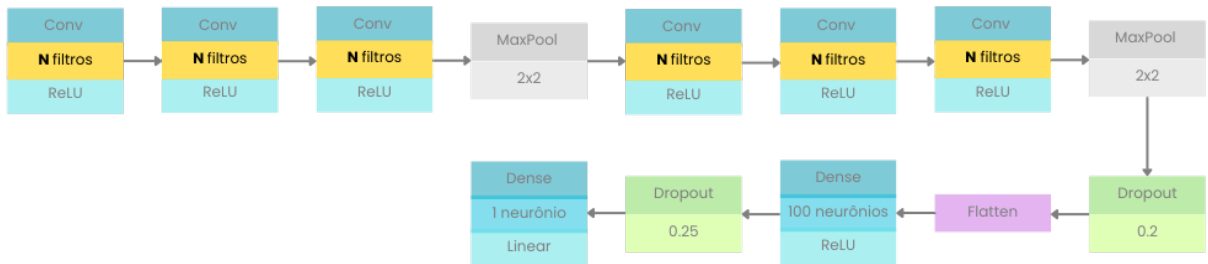


Figura 6.12 – Arquitetura base dos Modelos B. Os blocos destacados em amarelo representam as estruturas variadas em cada modelo.

## MODELO B.1

Nesta arquitetura cada camada convolucional foi implementada com 10 filtros. As curvas de função de perda e métrica ao longo das *epochs* para o treino e validação são apresentadas na figura 6.13, nos gráficos à esquerda e direita, respectivamente.

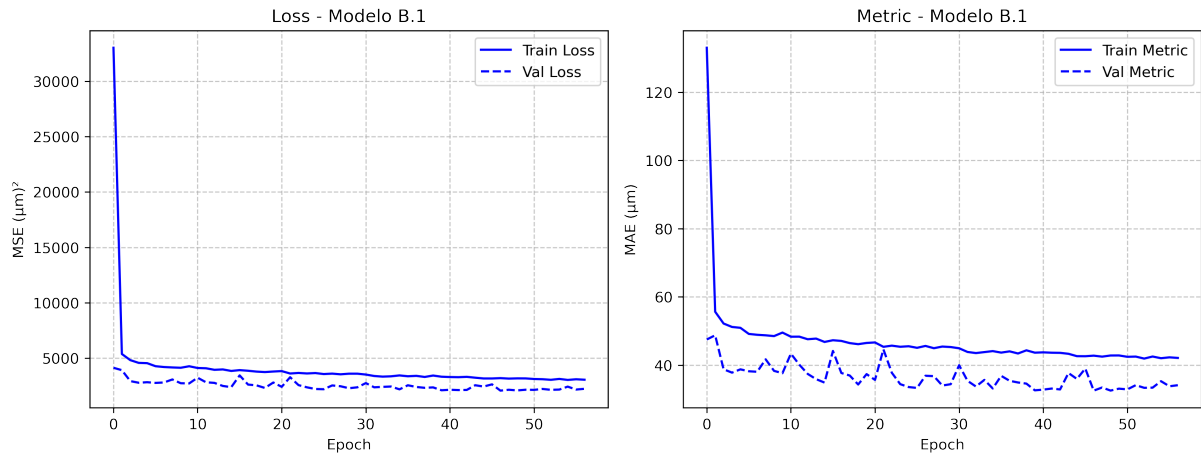


Figura 6.13 – Curvas de desempenho ao longo do treinamento e validação para o Modelo B.1.

A evolução das curvas de desempenho para o modelo B.1 mostra um pequeno *underfitting*, o que evidencia que os melhores resultados são obtidos durante a validação. O modelo converge com cerca de 55% das *epochs* disponíveis.

## MODELO B.2

Neste modelo as camadas convolucionais possuem 20 filtros. As curvas da função de perda e métrica em função das *epochs* estão reportadas na figura 6.14.

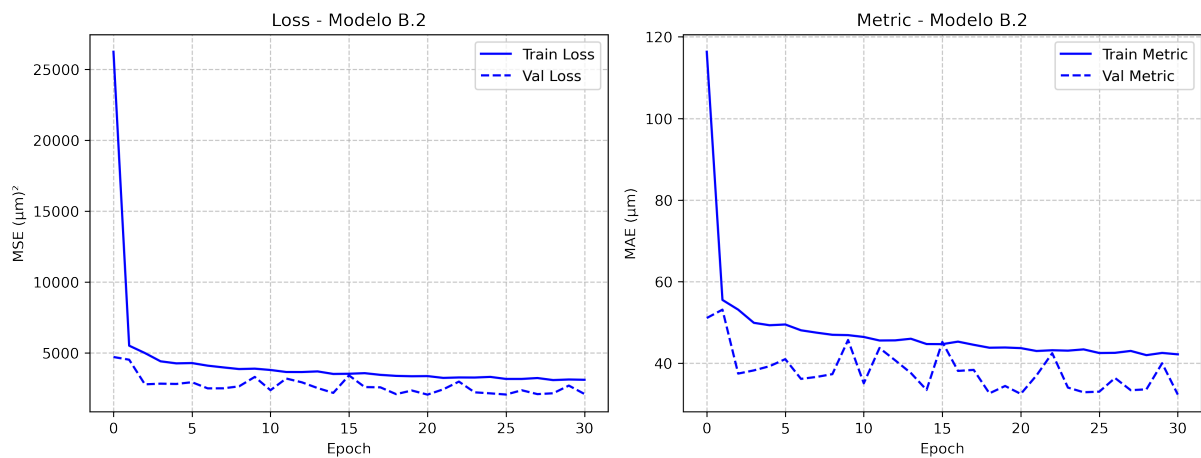


Figura 6.14 – Curvas de desempenho ao longo do treinamento e validação para o Modelo B.2.

O desempenho do modelo ao longo do treinamento e validação acusam que há *underfitting*. O modelo utiliza 30% das *epochs* para convergir.

### MODELO B.3

Nesta arquitetura cada camada convolucional possui 30 filtros. As curvas da função de perda e da métrica durante o treinamento e validação são mostradas na figura 6.15.

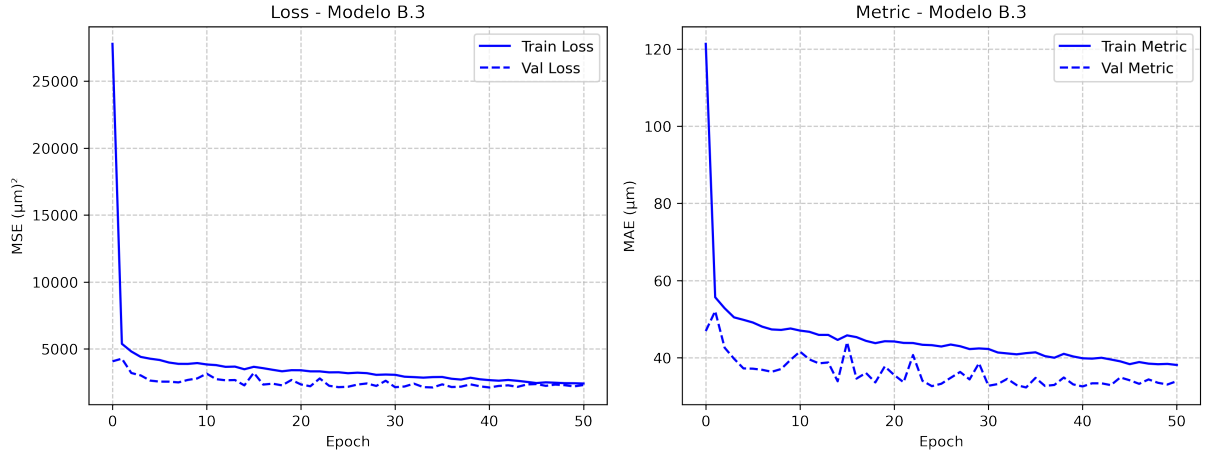


Figura 6.15 – Curvas de desempenho ao longo do treinamento e validação para o Modelo B.3.

Diferentemente da métrica, a curva da função de perda revela um *underfitting* no intervalo de 0 a 40 *epochs*, depois disto as curvas de treinamento e validação se encontram, o que não acontece no caso da função de perda. Este modelo utilizou 50% das *epochs* para convergir.

### MODELO B.4

Este modelo foi construído com 40 filtros em cada camada convolucional. As curvas das funções de perda e treinamento ao longo do treino e validação estão reportadas na figura 6.16.

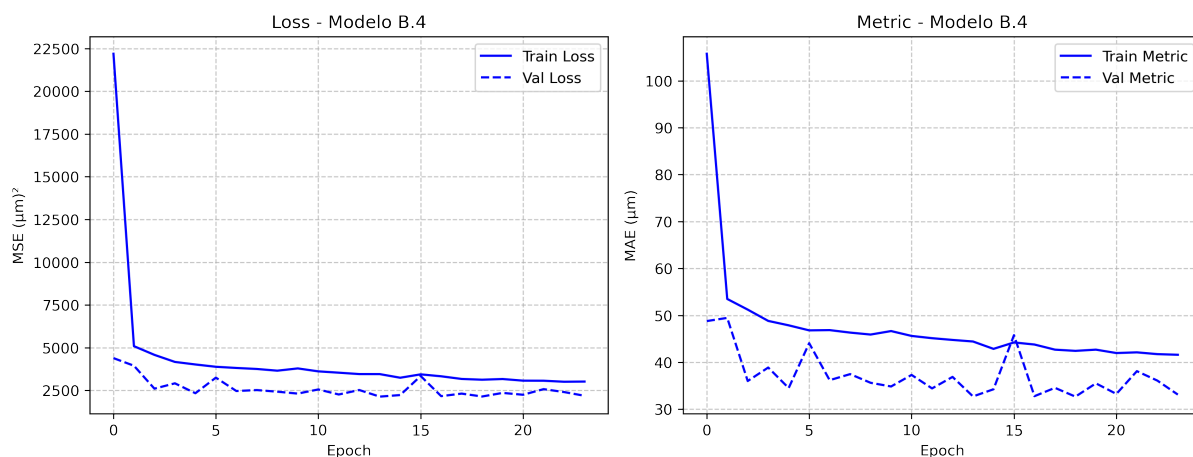


Figura 6.16 – Curvas de desempenho ao longo do treinamento e validação para o Modelo B.4.

A evolução das curvas de desempenho aponta para um caso de *underfitting*. Este modelo convergiu com menos *epochs* que os demais, utilizando pouco menos de 25% das iterações estipuladas.

#### COMPARAÇÃO ENTRE OS MODELOS

O desempenho das arquiteturas foi comparado através dos gráficos de perfil e resíduo, reportados na figura 6.17, e desvio padrão, que é apresentado na figura 6.18. Estes gráficos comparativos foram construídos da mesma forma que os da rodada anterior. Os valores que expressam o desempenho individual médio de cada modelo durante a validação estão listados na tabela 3.

Modelo	MSE $(\mu m)^2$	MAE $(\mu m)$
B.1	2069	32.51
B.2	2047	32.49
B.3	2122	32.53
B.4	2133	32.73

Tabela 3 – Valores da função de perda e métrica da validação para os modelos da rodada B.



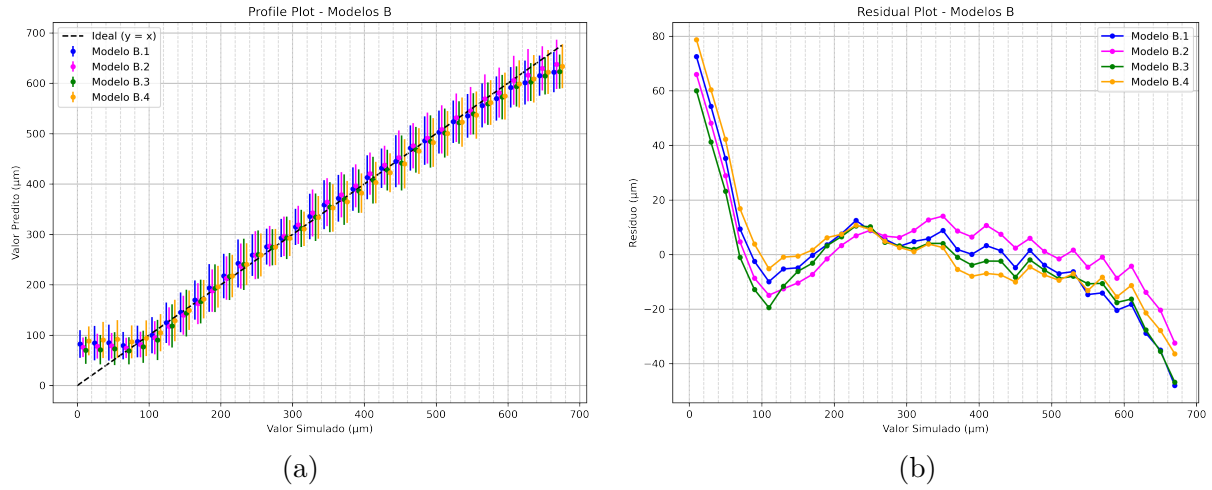


Figura 6.17 – *Profile Plot* (a) e *Residual Plot* (b) para os Modelos B.

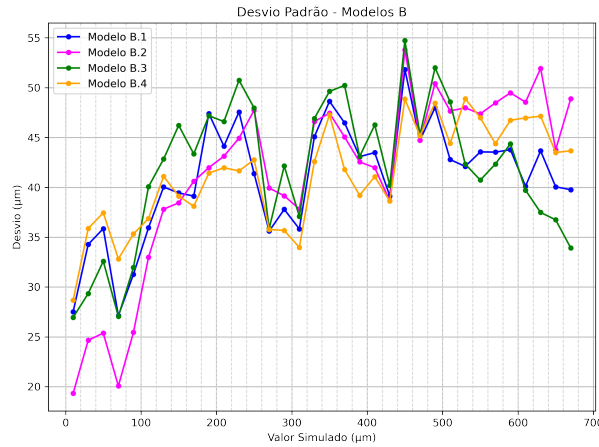


Figura 6.18 – Desvio padrão por *bin* para os Modelos B.

Os resultados do *profile* e *residual plot* nos revelam que no regime estrito de 300 a 500  $\mu\text{m}$  o modelo B.1 possui melhor desempenho, com menores valores para o resíduo. No entanto, para valores maiores que 500  $\mu\text{m}$ , o modelo B.2 é o que tem a melhor performance. O desvio padrão por *bin* mostra que B.2 possui menores devios na previsão das profundidades de eventos rasos, depois ele cresce até pouco menos de 50 m, enquanto os demais ficam abaixo de 45  $\mu\text{m}$ . Avaliando os gráficos comparativos e os resultados individuais de cada modelo ao longo do treino e validação, a arquitetura escolhida para a próxima rodada foi a B.2, com 20 filtros em cada camada convolucional. Este modelo também é o que possui menores valores de MSE e MAE.

### 6.3.3 MODELOS C

Este grupo de modelos explora o número ideal de neurônios na penúltima camada densa. Para este estudo, a arquitetura B.2, com 20 filtros por camada convolucional, foi preservada quanto aos outros hiperparâmetros. O diagrama 6.19 representa a arquitetura

básica dos modelos desta rodada, e o bloco destacado em amarelo mostra o hiperparâmetro que é alterado em cada caso.

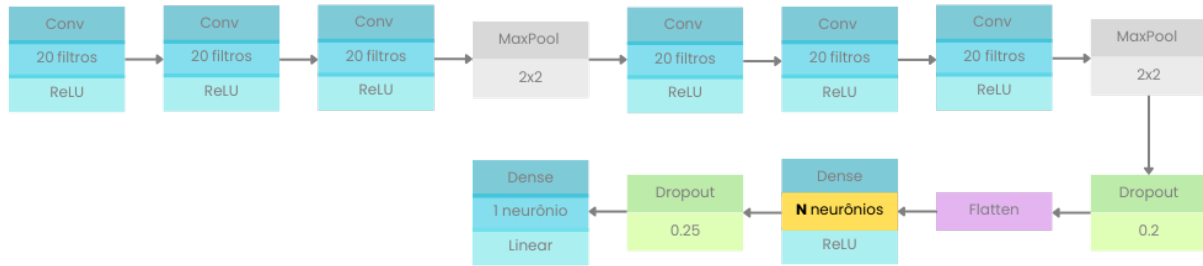


Figura 6.19 – Arquitetura base dos Modelos C. O bloco em amarelo destaca o hiperparâmetro variado nos modelos.

### MODELO C.1

Nesta arquitetura a penúltima camada densa possui 10 neurônios. As curvas das funções de perda e métrica ao longo das *epochs* durante o treinamento e validação estão reportadas na figura 6.20.

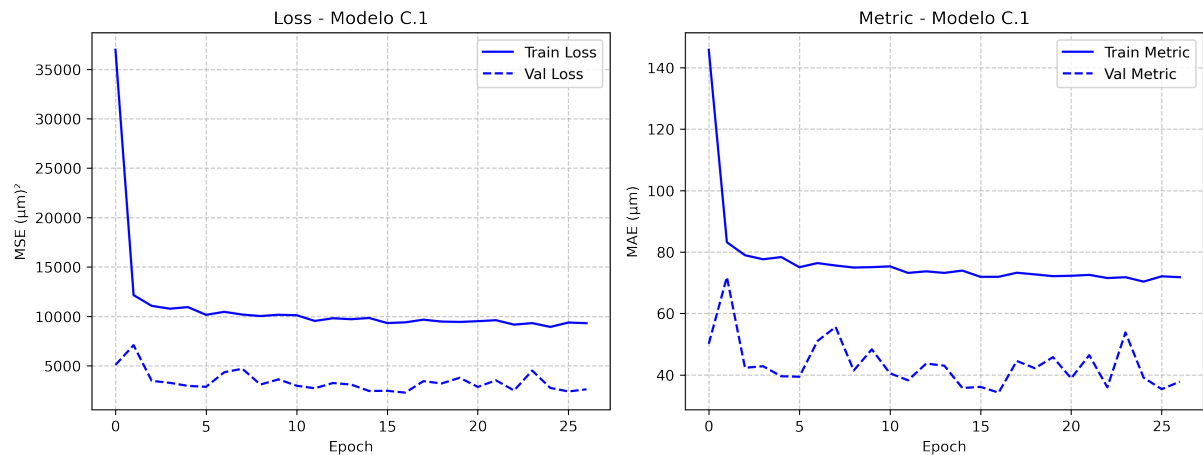


Figura 6.20 – Curvas de desempenho ao longo do treinamento e validação para o Modelo C.1.

As curvas de desempenho indicam um *underfitting* mais acentuado que os vistos anteriormente, nos modelos de outras rodadas. Isto é evidenciado na distância entre as curvas de treinamento e validação. A baixa complexidade deste modelo está relacionada aos poucos neurônios utilizados na penúltima camada densa. A convergência para esta arquitetura se deu com pouco mais de 25% das *epochs*.

### MODELO C.2

Este modelo possui 50 neurônios na penúltima camada densa. As curvas das funções de perda e métrica por *epochs* são apresentadas na figura 6.21.

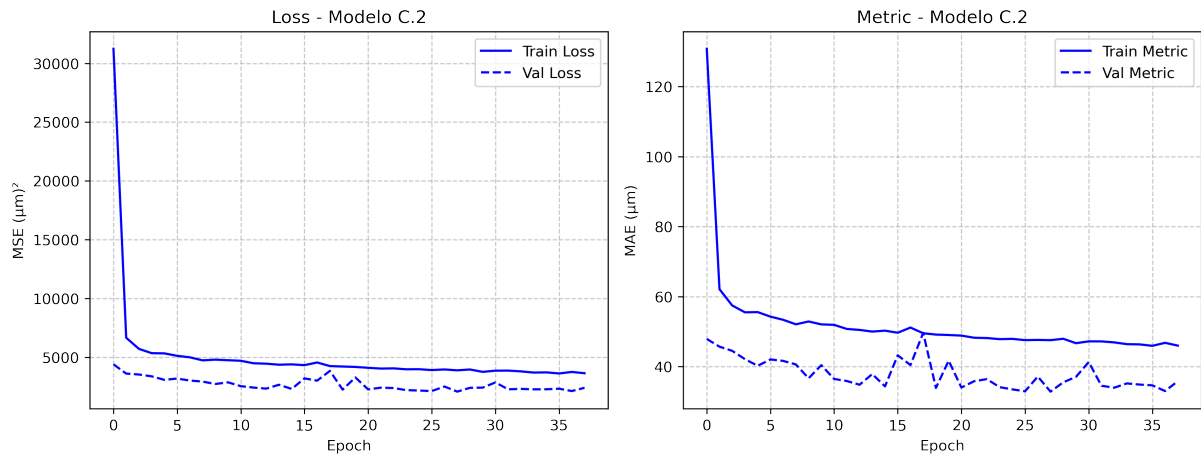


Figura 6.21 – Curvas de desempenho ao longo do treinamento e validação para o Modelo C.2.

Os gráficos apontam um *underfitting* menor que os observados no modelo C.1, mostrado anteriormente. Com a presente arquitetura a convergência se deu com mais de 35% das *epochs*.

### MODELO C.3

Este modelo foi implementado com 100 neurônios na penúltima camada densa. As curvas de perda e métrica por *epochs* são apresentadas na figura 6.22.

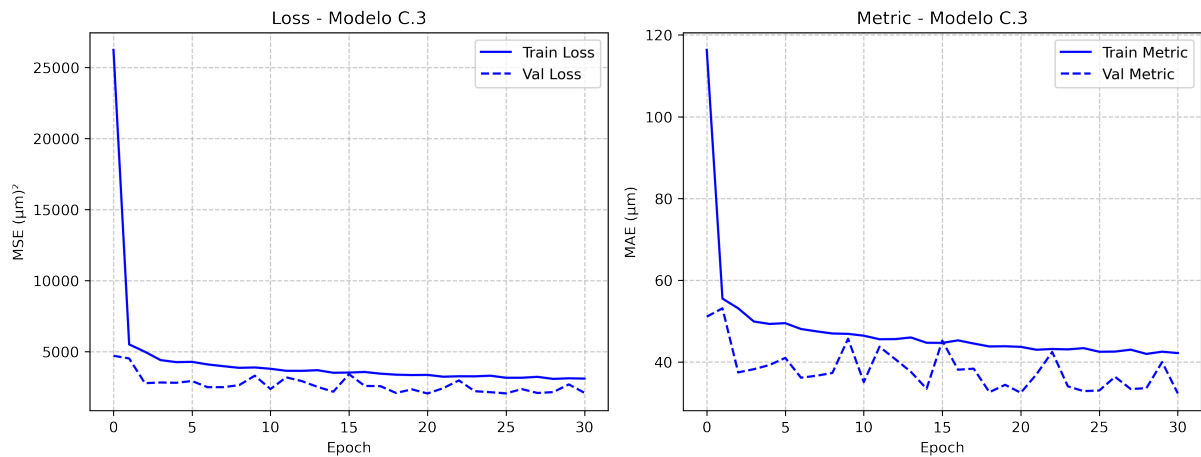


Figura 6.22 – Curvas de desempenho ao longo do treinamento e validação para o Modelo C.3.

O desempenho do modelo ao longo do treinamento e da validação revela *underfitting*. A convergência ocorre com 30% das *epochs* estipuladas.

### MODELO C.4

A penúltima camada densa deste modelo possui 200 neurônios. As funções de perda e métrica ao longo das *epochs* são representadas pelas curvas dos gráficos na figura 6.23.

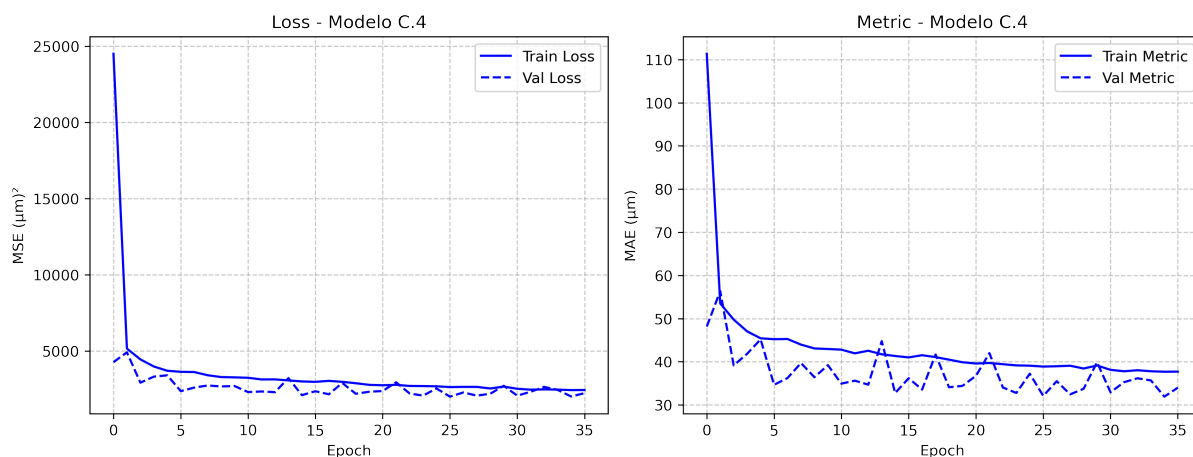


Figura 6.23 – Curvas de desempenho ao longo do treinamento e validação para o Modelo C.4.

A evolução das curvas de treinamento e validação para este modelo, mostram o menor *underfitting* em relação as demais arquiteturas desta rodada. Depois de 25 iterações, as curvas da função de perda coincidem. O modelo utiliza 35% das *epochs* para alcançar a convergência.

#### COMPARAÇÃO DOS MODELOS

A comparação entre os quatro modelos do grupo C se deu através dos gráficos de *profile* e *residual plot*, reportados na figura 6.24, e do desvio padrão por *bin*, que é apresentado na figura 6.25. Os desempenhos individuais de cada modelo durante a validação são reportados na tabela 4.

Modelo	MSE ( $\mu m$ ) <sup>2</sup>	MAE ( $\mu m$ )
C.1	2275	34.22
C.2	2079	32.87
C.3	2047	32.49
C.4	2005	32.09

Tabela 4 – Valores da função de perda e métrica da validação para os modelos da rodada C.

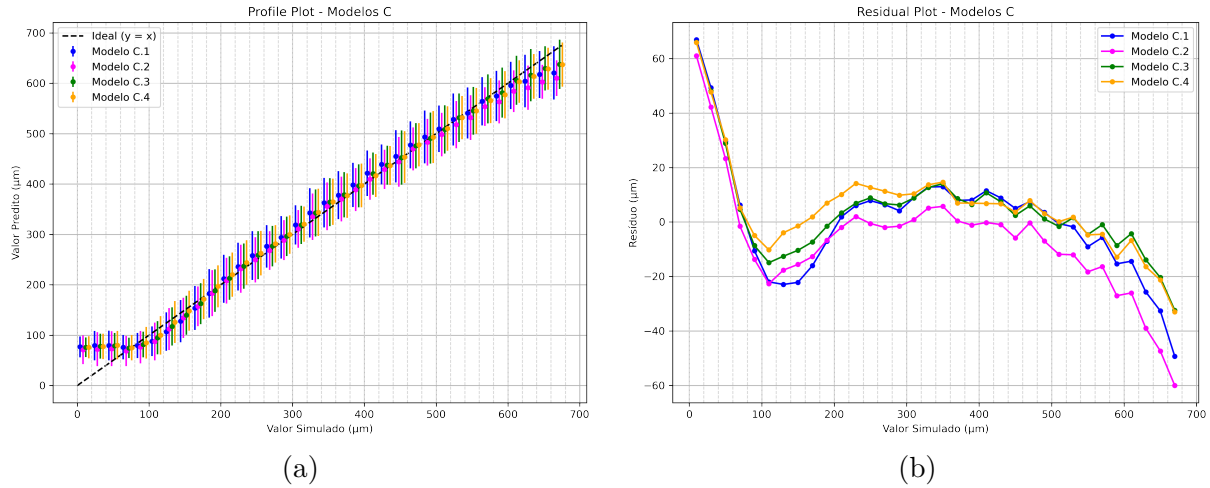


Figura 6.24 – *Profile Plot* (a) e *Residual Plot* (b) para os Modelos C.

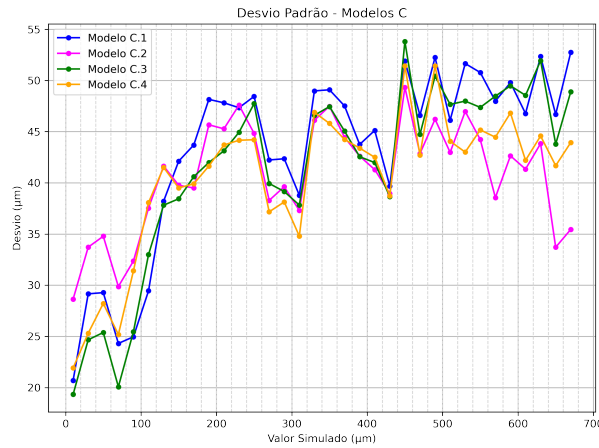


Figura 6.25 – Desvio padrão por *bin* para os Modelos C.

Observando o *profile* e o *residual plot*, é nítido que o modelo C.2 apresenta melhores resultados no regime de 200 a 500  $\mu\text{m}$ , enquanto os demais têm desempenhos parecidos. Mas isso muda em profundidades maiores, onde os modelos C.3 e C.4 tem melhores performances. O desvio padrão por *bin* indica que todos os modelos tem dispersão parecida em predições no intervalo de 100 a 500  $\mu\text{m}$ , depois disso o modelo C.2 atinge a menor dispersão máxima dentre os outros. Antes de fazer a escolha da melhor arquitetura, faremos a validação cruzada, técnica apresentada no capítulo anterior, para entender como a performance destes modelos depende do conjunto de dados.

#### 6.3.4 K-FOLD CROSS VALIDATION

Para estudar como a performance dos modelos C depende dos conjuntos de dados utilizados para o treinamento implementamos a técnica de validação cruzada, comentada anteriormente no capítulo 5, que testa a capacidade de generalização do modelo para

dados não vistos. Em todos os casos o *dataset* destinado a treino e validação foi dividido em 5 *folds* de mesmo tamanho.

### CROSS VALIDATION PARA O MODELO C.1 E COMPARAÇÃO DOS DIFERENTES FOLDS

Cada um dos *folds* utilizados no treinamento e validação é um modelo independente, com suas próprias curvas de perda e métrica ao longo das *epochs*. A validação cruzada foi feita com o modelo C.1, com 10 neurônios na última camada densa oculta. Estas curvas estão apresentadas nos gráficos da figura 6.26 e mostram uma discrepância considerável entre cada *fold*. O *fold* 1 é o que tem melhores resultados tanto para função de perda quanto para a métrica. Os modelos convergiram com pouco menos de 50% das *epochs*.

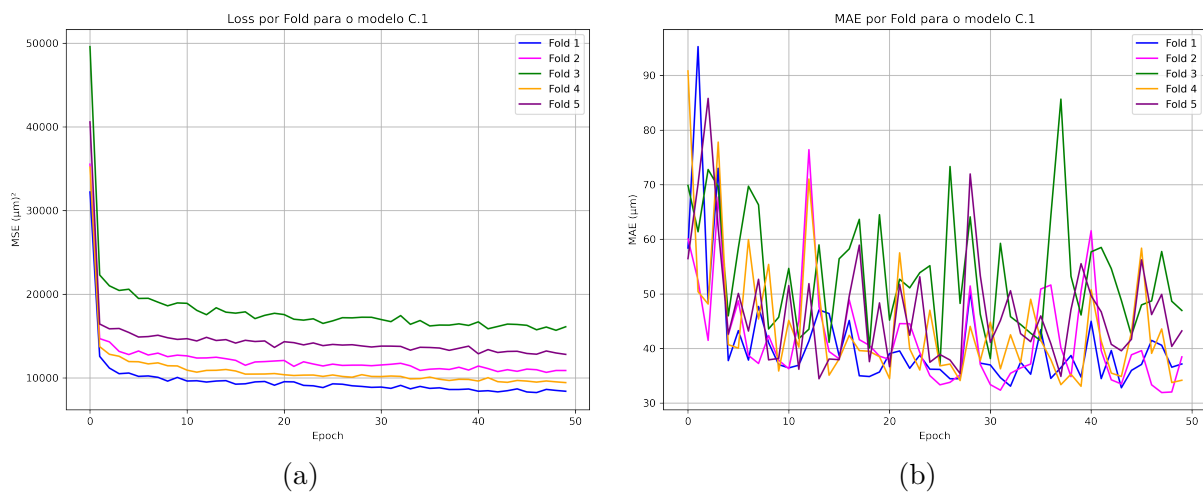


Figura 6.26 – Curvas das funções de *loss* (a) e métrica (b) por *fold* para o Modelo C.1.

Para comparar as predições feitas por cada um dos *folds* sobre o conjunto de testes utilizamos os gráficos de *profile* e *residual plot*, apresentados na figura 6.27. A análise destas curvas nos mostra que o *fold* 4 tem melhor performance na predição de valores no regime entre 100 e 300  $\mu\text{m}$ , e depois passa a se distanciar, de forma acentuada, do comportamento ideal (resíduo igual a zero), da mesma forma que os demais.

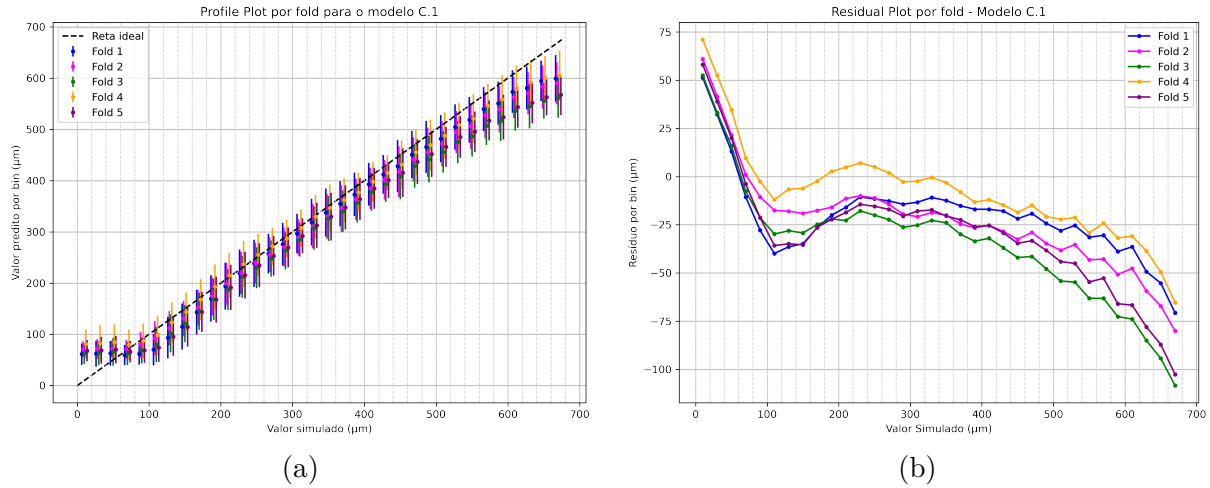


Figura 6.27 – *Profile Plot* (a) e *Residual Plot* (b) por *fold* para o Modelo C.1.

Os valores individuais das funções de perda e métrica durante a validação estão descritos na tabela 5, bem como, o desempenho médio dos *folds*.

<i>Fold</i>	MSE ( $\mu m$ ) <sup>2</sup>	MAE ( $\mu m$ )
1	2559	37
2	2636	38
3	3585	47
4	2315	34
5	3165	43
Médias	$2852 \pm 459$	$40 \pm 4$

Tabela 5 – Valores da função de perda e métrica para diferentes *folds* para o modelo C.1 .

Analisando os desempenhos individuais para cada um dos *folds* percebemos que os valores mudam consideravelmente de um *fold* para outro. Isto é evidenciado no desvio padrão associado à distribuição dos valores da função de perda, igual a  $459 \mu m^2$ .

#### CROSS VALIDATION PARA O MODELO C.2 E COMPARAÇÃO DOS DIFERENTES *FOLDS*

Fizemos a validação cruzada para o modelo C.2, com 50 neurônios na última camada densa oculta. As curvas da perda e métrica para cada *fold* são apresentadas nos gráficos da figura 6.28. Avaliando a evolução destas curvas podemos notar que elas têm comportamento similar e mais uniforme entre si, bem diferente do observado na validação cruzada do modelo anterior.

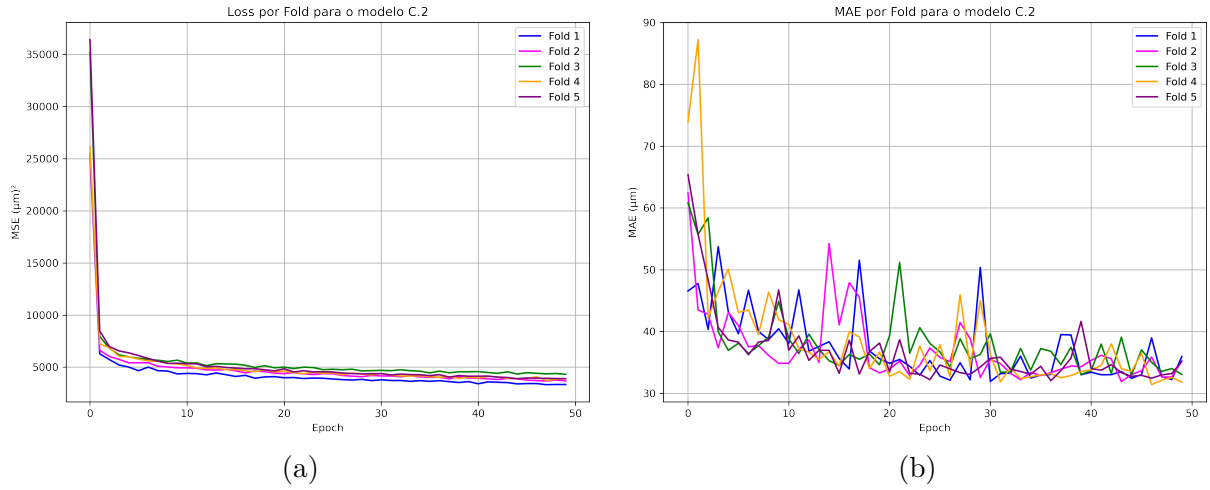


Figura 6.28 – Curvas das funções de *loss* (a) e métrica (b) por *fold* para o Modelo C.2.

Através de *profile* e do *residual plot*, reportados na figura 6.29, comparamos as capacidades preditivas dos diferentes *fold*s.

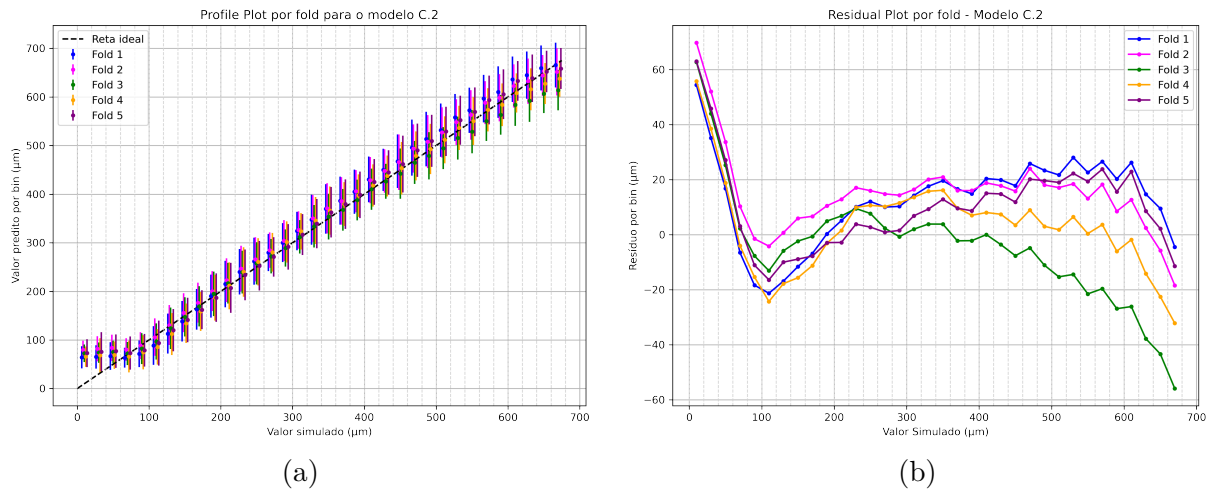


Figura 6.29 – *Profile Plot* (a) e *Residual Plot* (b) por *fold* para o Modelo C.2.

Os valores individuais das funções de perda e métrica estão reportados na tabela 6.

<i>Fold</i>	MSE ( $\mu m$ ) <sup>2</sup>	MAE ( $\mu m$ )
1	2454	36
2	2326	35
3	2109	33
4	1988	32
5	2362	35
Médias	$2248 \pm 172$	$34 \pm 2$

Tabela 6 – Valores da função de perda e métrica para diferentes *fold*s para o modelo C.2 .

No caso da função de perda, o desvio padrão da distribuição de valores é de  $172 \mu m^2$ , bem menor que obtido na validação cruzada do modelo antecessor.



## CROSS VALIDATION PARA O MODELO C.3 E COMPARAÇÃO DOS DIFERENTES *FOLDS*

A validação cruzada foi feita para o modelo C.3, com 100 neurônios na última camada densa oculta. Observando a evolução da função de perda para os diferentes *folds*, apresentada na figura 6.30 (a), percebemos que as curvas são praticamente coincidentes. Isto revela uma uniformidade no processo de validação, que também é percebida nas curvas de métrica, reportadas na figura 6.30 (b).

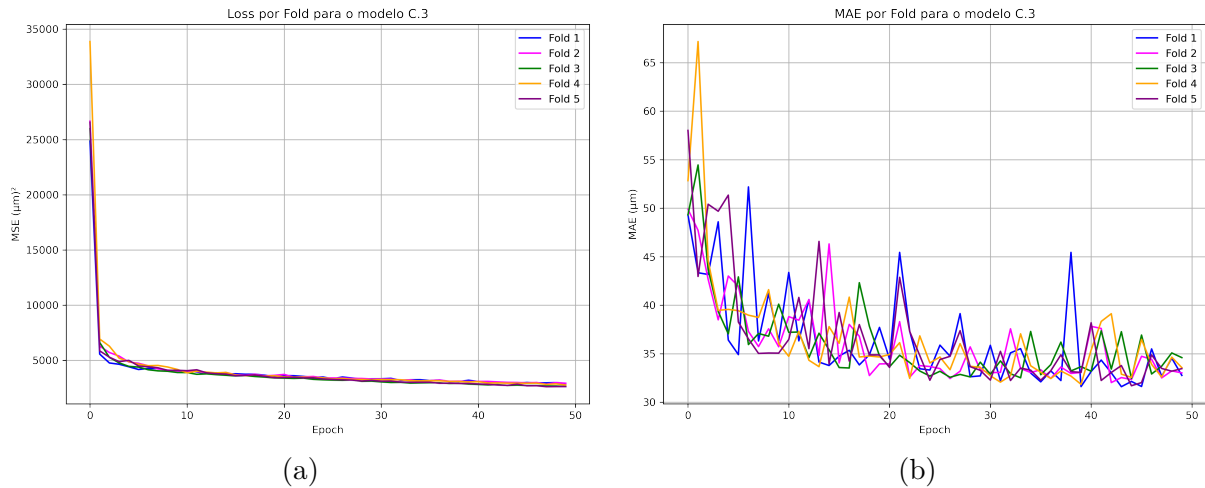


Figura 6.30 – Curvas das funções de *loss* (a) e métrica (b) por *fold* para o Modelo C.3.

Para comparar a eficiência preditiva para cada *fold* utilizamos o *profile plot* e o *residual plot*, que são mostrados na figura 6.31.

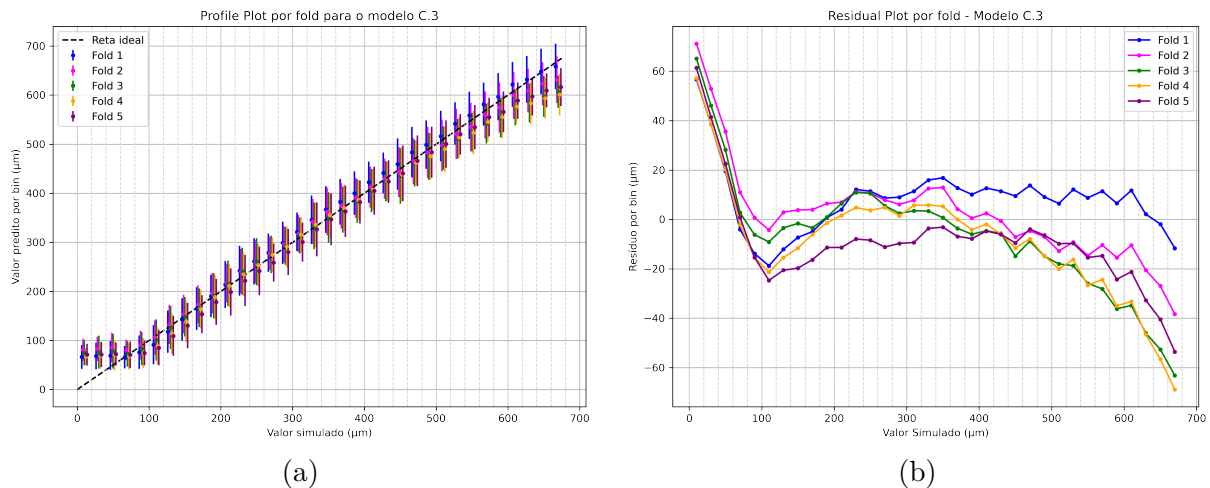


Figura 6.31 – *Profile Plot* (a) e *Residual Plot* (b) por *fold* para o Modelo C.3.

Na tabela 7 estão descritos os valores individuais das funções de perda e métrica para cada *fold*. Também constam os valores médios obtidos. O desvio padrão das distribuições para os dois casos evidenciam que a validação é estável, independente dos *folds*.

<i>Fold</i>	MSE ( $\mu m$ ) <sup>2</sup>	MAE ( $\mu m$ )
1	2182	32.7
2	2143	33.0
3	2286	34.6
4	2265	33.5
5	2145	33.4
Médias	2204 $\pm$ 60	33.5 $\pm$ 0.6

Tabela 7 – Valores da função de perda e métrica para diferentes *folds* para o modelo C.3 .

#### CROSS VALIDATION PARA O MODELO C.4 E COMPARAÇÃO DOS DIFERENTES *FOLDS*

Fizemos a validação cruzada para o modelo C.4, com 200 neurônios na última camada densa oculta. As curvas das funções de perda e métrica por *fold* são reportadas na figura 6.32. A evolução da perda ao longo das *epochs* revela uma validação bem coesa entre os *folds*, com curvas coincidentes. O mesmo pode ser observado na dinâmica das curvas de métrica, que são, em média, estáveis ao longo da validação.

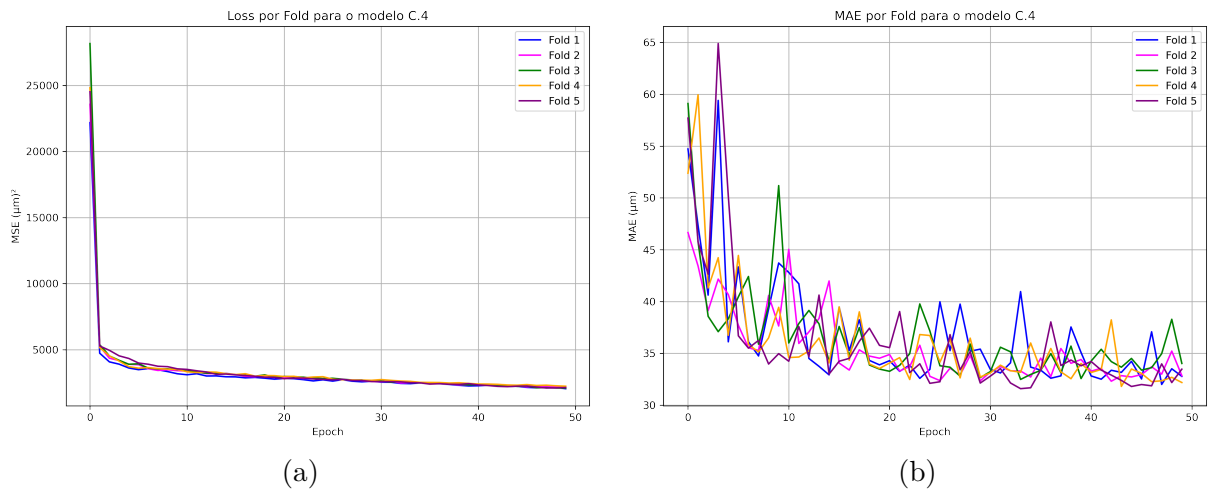


Figura 6.32 – Curvas das funções de *loss* (a) e métrica (b) por *fold* para o Modelo C.4.

A comparação das predições em cada *fold* foi feita através do *profile plot* e do *residual plot*, reportados na figura 6.33.

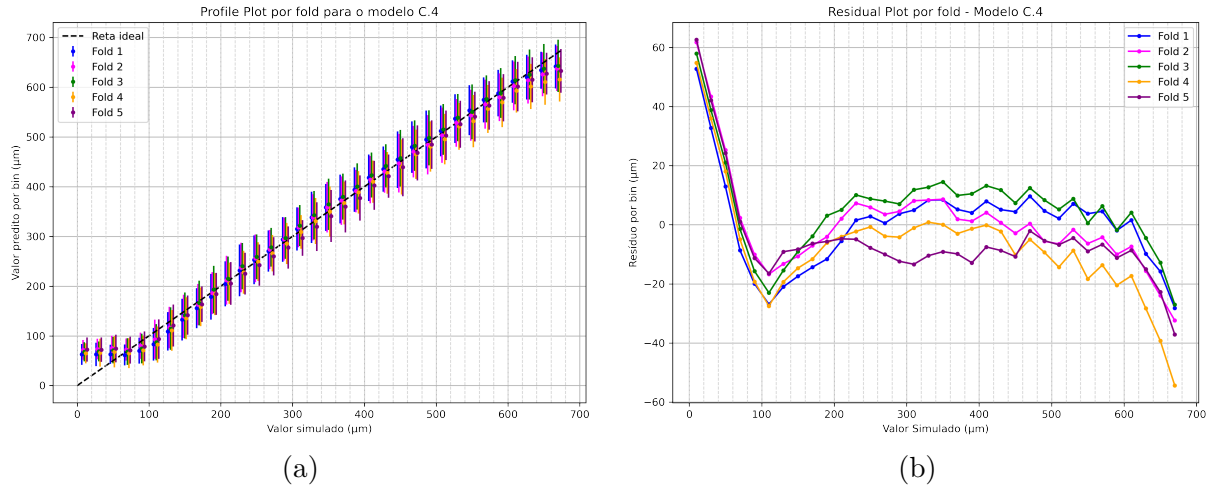


Figura 6.33 – *Profile Plot* (a) e *Residual Plot* (b) por *fold* para o Modelo C.4.

Os valores individuais das funções de perda e métrica para cada *fold* estão descritos na tabela 8, assim como as média para as duas distribuições. Os baixos valores obtidos para o desvio padrão são fortes evidências de que a performance da cada *fold* durante a validação é bem uniforme.

<i>Fold</i>	MSE ( $\mu m$ ) <sup>2</sup>	MAE ( $\mu m$ )
1	2268	32.7
2	2256	32.8
3	2190	34.0
4	2111	32.2
5	2171	33.5
Médias	$2200 \pm 58$	$33.0 \pm 0.6$

Tabela 8 – Valores da função de perda e métrica para diferentes *folds* para o modelo C.4 .

A validação cruzada nos mostra que os modelos C.3 e C.4 apresentam performances bem mais uniformes por *folds*. Os resultados obtidos para estas duas arquiteturas são equivalentes, com valores médios de MSE e MAE compatíveis levando em conta as dispersões. Ambos os modelos tem capacidades preditivas similares em regimes de maiores profundidades. Mas neste primeiro momento, o modelo C.3 é mais atrativo pois possui menos neurônios (100) na camada totalmente conectada e por isso tem menos custo computacional. No entanto, ainda faremos mais um estudo, que será detalhado na seção seguinte, para entender a influência da adição de uma camada intermediária totalmente conectada na performance dos modelos C. A partir destes resultados vamos determinar o melhor modelo.

### 6.3.5 MODELOS D

Nesta rodada estudamos a influência de uma camada densa intermediária entre o último *dropout* e a camada de saída. Para isto utilizamos as arquiteturas dos modelos C e

adicionamos em cada rede uma camada oculta com 5 neurônios. O diagrama da figura 6.34 apresenta a arquitetura comum a todos as arquiteturas deste grupo, em amarelo temos os hiperparâmetros que mudam em cada um dos casos.

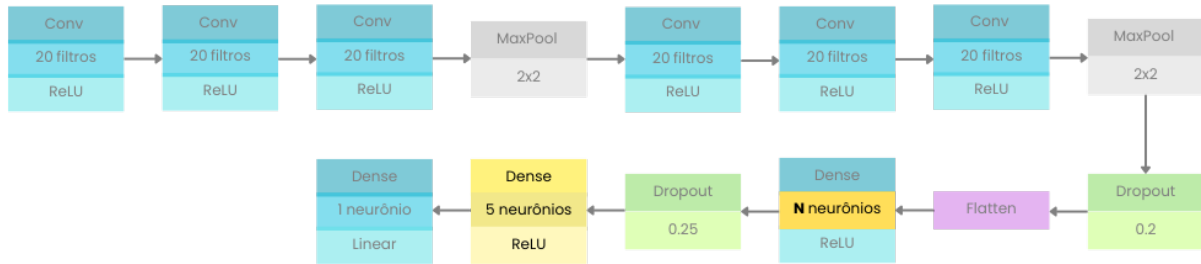


Figura 6.34 – Arquitetura base dos Modelos C. Os blocos em amarelo representam as estruturas variadas nos modelos.

### MODELO D.1

Este modelo conta com 10 neurônios na penúltima camada densa e 1 camada intermediária com 5 neurônios. As curvas das funções de perda e métrica ao longo das *epochs* para o treinamento e validação são apresentadas na figura 6.35.

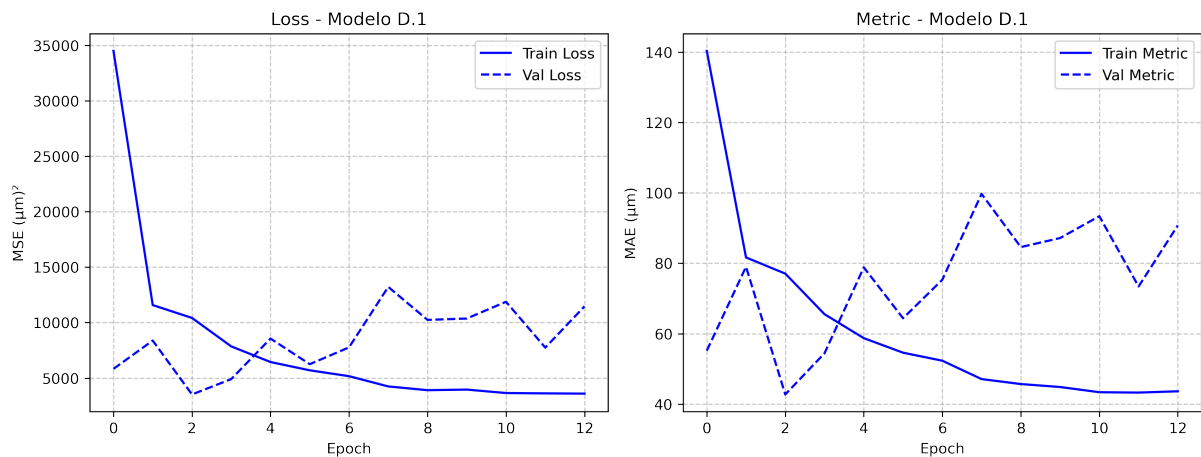


Figura 6.35 – Curvas de desempenho ao longo do treinamento e validação para o Modelo D.1.

A evolução das curvas mostra um caso de *overfitting*, evidenciado pelo distanciamento das curvas de treino e validação e dos melhores resultados durante o treinamento a partir da terceira *epoch*. O modelo converge com 12 % das *epochs* disponíveis.

### MODELO D.2

Este modelo conta com duas camadas densas, de 50 e 5 neurônios. As curvas da função de perda e métrica ao longo das *epochs* durante o treino e a validação são apresentadas na figura 6.36.

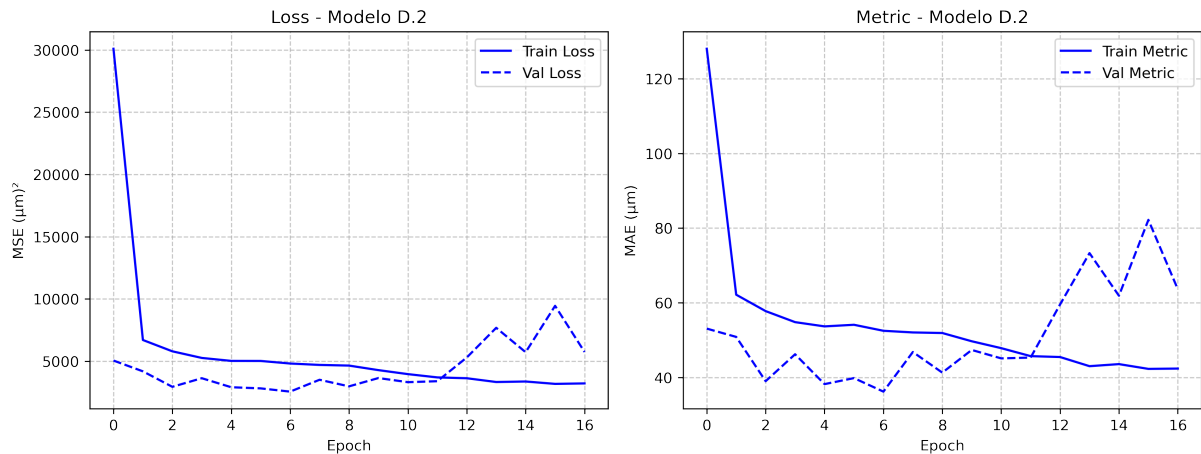


Figura 6.36 – Curvas de desempenho ao longo do treinamento e validação para o Modelo D.2.

A evolução das curvas mostra que há *overfitting*. A distância entre as curvas de treinamento e validação para as funções de perda e métrica são menores do que as reportadas no modelo anterior, D.1, diferido pelo uso de 10 neurônios na primeira camada densa.

### MODELO D.3

O modelo D.3 possui duas camadas densas ocultas, com 100 e 5 neurônios. As curvas da função de perda e métrica ao longo das *epochs* são apresentadas na figura 6.37.

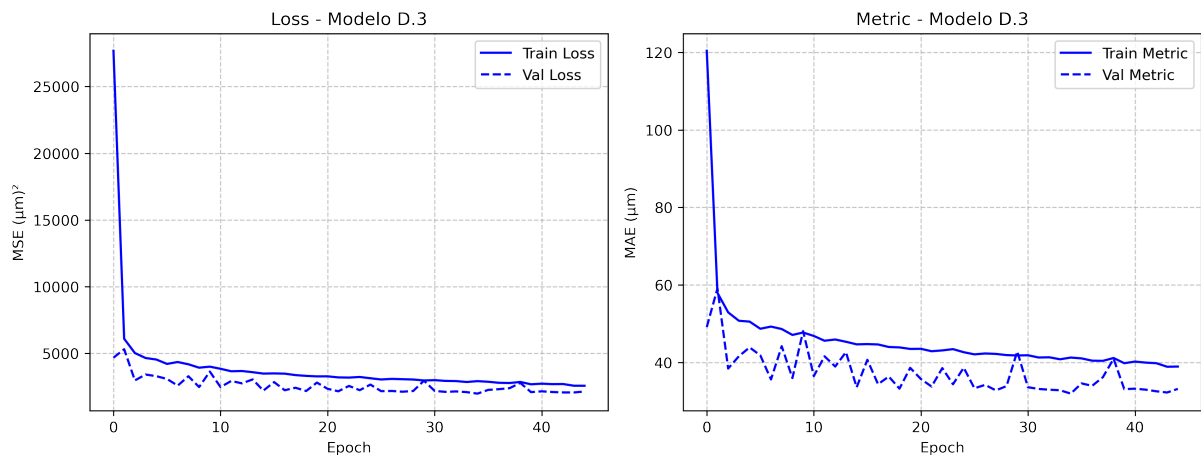


Figura 6.37 – Curvas de desempenho ao longo do treinamento e validação para o Modelo D.3.

As curvas de desempenho indicam um *underfitting* bem pequeno. No caso da função de perda, as curvas de treinamento e validação são distanciadas por bem pouco. O modelo atingiu a convergência com pouco mais de 40% das *epochs* estipuladas.

## MODELO D.4

O modelo D.4 combina duas camadas densas ocultas com 200 e 5 neurônios. As curvas das funções de perda e métrica ao longo das *epochs* estão registradas na figura 6.38.

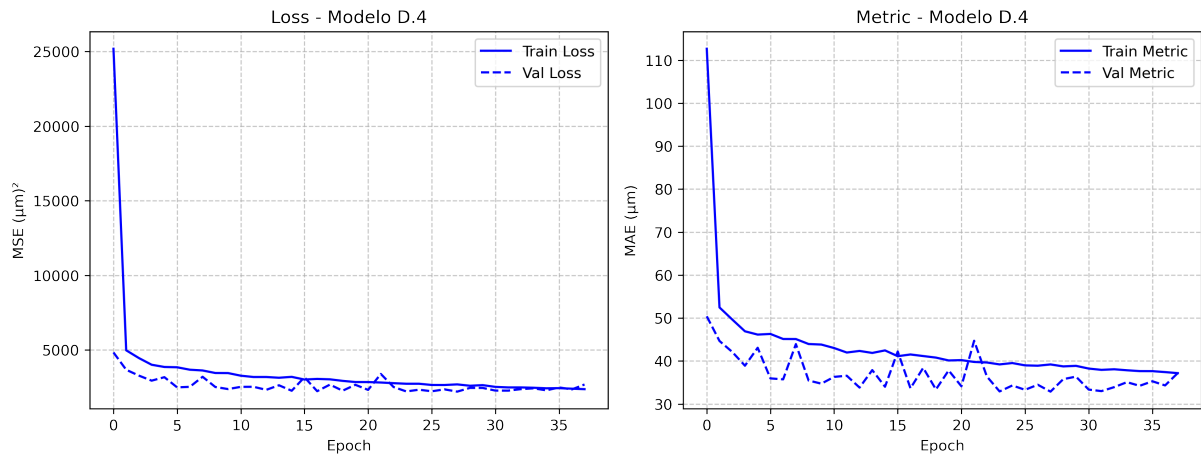


Figura 6.38 – Curvas de desempenho ao longo do treinamento e validação para o Modelo D.4.

A evolução das curvas de treinamento e validação aponta para um *underfitting* pequeno no caso da função de perda, que ocorre ao longo das *epochs* mas é atenuado nas iterações finais. O modelo usou pouco mais de 35% das *epochs* para convergir.

## COMPARAÇÃO ENTRE OS MELHORES MODELOS DA RODADA C E D.

Analisando as curvas de treinamento e validação para cada modelo individualmente, nota-se que as redes D.3 e D.4 têm a melhor performance. Isto também é percebido na tabela 9, que lista os valores médios da perda e da métrica durante a validação.

Modelo	MSE $(\mu m)^2$	MAE $(\mu m)$
D.1	3520	42.74
D.2	2565	36.18
D.3	1988	31.94
D.4	2197	32.86

Tabela 9 – Valores da função de perda e métrica da validação para os modelos da rodada D.

Para entendermos se a adição de uma camada densa intermediária é benéfica para o desempenho destas arquiteturas vamos compará-las com as que tiveram o melhor resultado na rodada C (C.3 e C.4). Os gráficos de *profile* e *residual plot* para estes quatro modelos são apresentados na figura 6.39. O desvio padrão por *bin* é mostrado na figura 6.40.

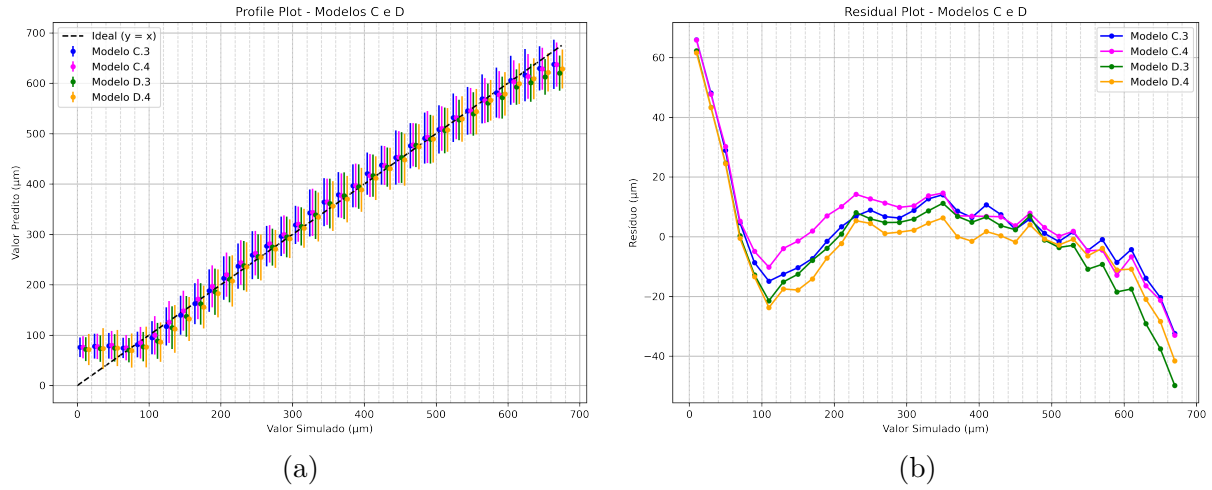


Figura 6.39 – *Profile Plot* (a) e *Residual Plot* (b) para os melhores modelos das rodadas C e D.

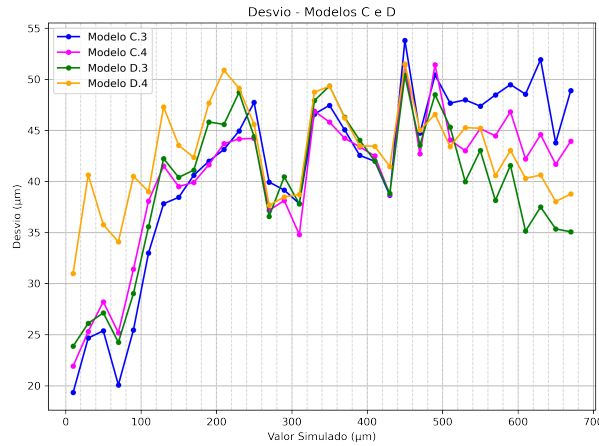


Figura 6.40 – Desvio padrão por *bin* para os melhores modelos das rodadas C e D.

Com a avaliação conjunta do *profile* e *residual plot* concluímos que a camada densa intermediária é vantajosa para um regime de 200 a 500  $\mu m$ , já que neste intervalos os modelos D.3 e D.4 possuem os menores valores de resíduo. Este comportamento, no entanto, não é preservado em profundidades maiores, onde os modelos C.3 e C.4 têm performance melhores. O desvio padrão por *bin* mostra que os modelos C.3, C.4 e D.3 têm as menores dispersões na predição de eventos rasos, com profundidades menores que 100  $\mu m$ . No regime de 200 a 500  $\mu m$  esses quatro modelos apresentam comportamentos parecidos. Mas em profundidades maiores que isto, o modelo D.3 apresenta menor desvio padrão. Numa avaliação conjunta dos resultados obtidos para este grupo e do desempenho dos modelos na validação cruzada, descrita na rodada anterior, escolhemos o modelo C.3, com uma única camada densa oculta de 100 neurônios, para seguir na próxima rodada. Este modelo é o que possui a combinação de um desvio padrão e um viés menor na média das profundidades.

### 6.3.6 MODELOS E

Nesta rodada exploramos diferentes combinações das taxas de *dropout*. O modelo C.3, cuja a arquitetura é representada no diagrama 6.41, foi utilizado como base para este estudo, em amarelo estão destacadas as duas camadas de *dropout* que tiveram as taxas modificadas.

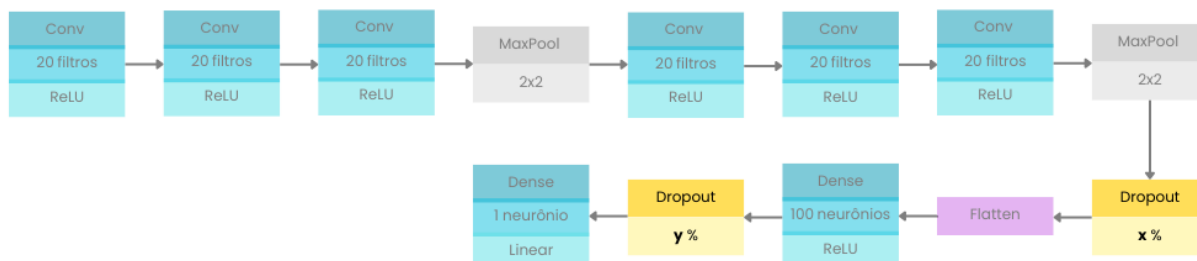


Figura 6.41 – Arquitetura base dos Modelos E. Estão destacados em amarelo os blocos de *dropout*, cujas taxas são variadas em cada modelo.

#### MODELO E.1

Nesta arquitetura as duas camadas de *dropout*, possuem taxas x e y iguais a 0.1. As curvas da função de perda e da métrica ao longo das *epochs* são apresentadas nos gráficos da figura 6.42.

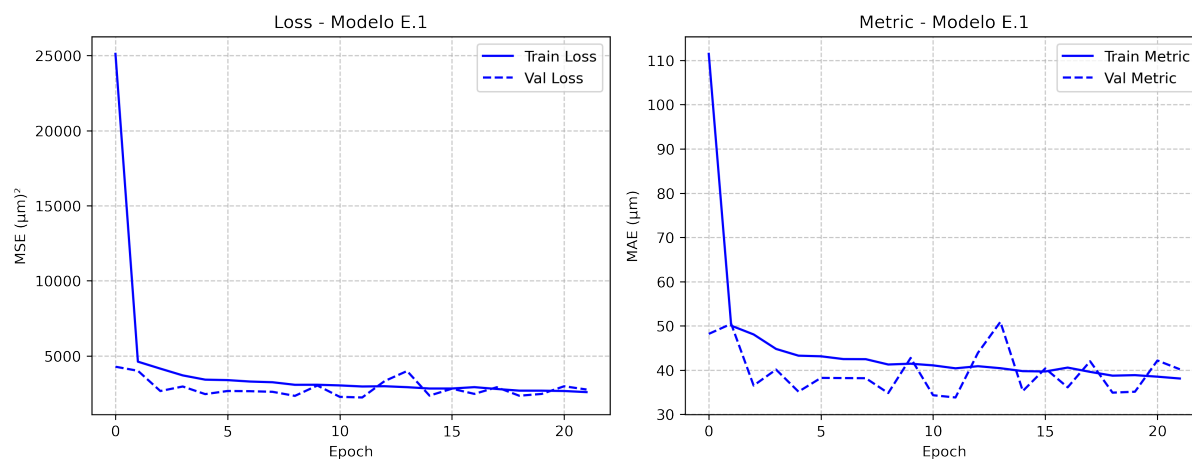


Figura 6.42 – Curvas de desempenho ao longo do treinamento e validação para o Modelo E.1.

A evolução da curva da função de perda aponta para uma relação estável de desempenho no treinamento e validação. Estes resultados nos mostram que a redução das taxas de *dropout*, em comparação as utilizadas nos modelos até então (0.2 e 0.25), implica na atenuação do *underfitting*. O modelo E.1 utilizou pouco mais de 20% das *epochs* até atingir a convergência.



## MODELO E.2

Para este modelo combinamos camadas de *dropout* com taxas iguais a 0.1 e 0.15. As curvas das funções de perda e métrica ao longo das *epochs* são reportadas nos gráficos da figura 6.43.

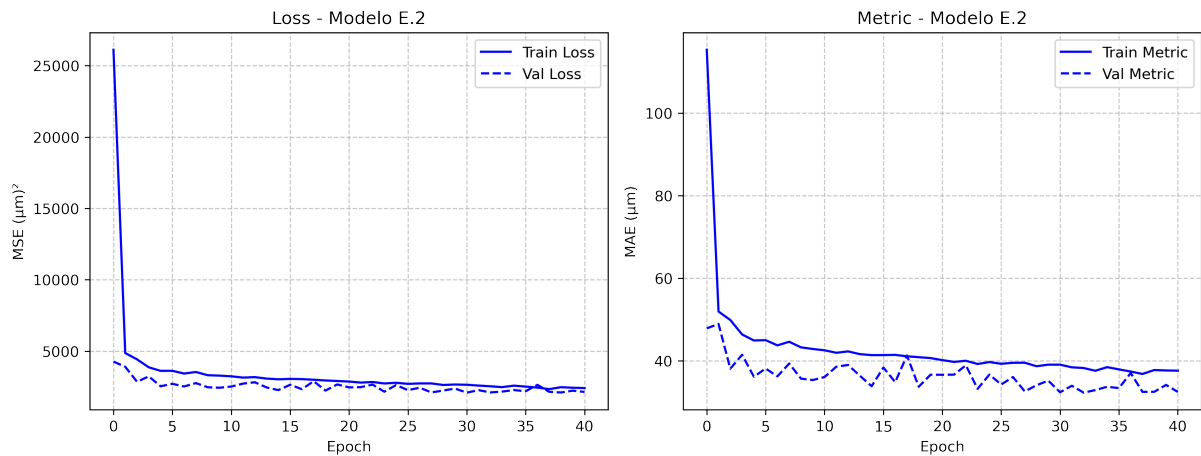


Figura 6.43 – Curvas de desempenho ao longo do treinamento e validação para o Modelo E.2.

A dinâmica das curvas treino e validação indicam um desempenho estável para a função de perda. No caso da métrica há um pequeno *underfitting*. Este modelo atingiu a convergência com 40% das *epochs*.

## MODELO E.3

Neste modelo as duas camadas de *dropout* possuem taxas iguais a 0.2. As curvas de perda e métrica ao longo das *epochs* para o treinamento e validação são mostradas na figura 6.44.

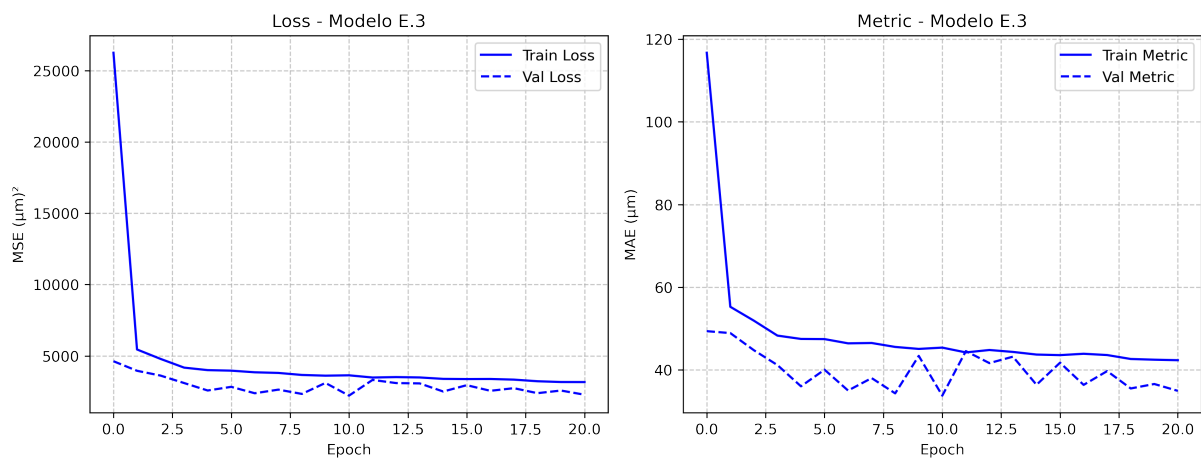


Figura 6.44 – Curvas de desempenho ao longo do treinamento e validação para o Modelo E.3.

A evolução das curvas de treino e validação para a função de perda indicam um *underfitting* bem tênue. O modelo convergiu utilizando 20% das *epochs*.

#### MODELO E.4

Neste modelo as camadas de *dropout* possuem taxas iguais a 0.2 e 0.25, as mesmas utilizadas nas arquiteturas descritas nas rodadas anteriores. As curvas das funções de perda e validação ao longo das *epochs* são apresentadas nos gráficos da figura 6.45.

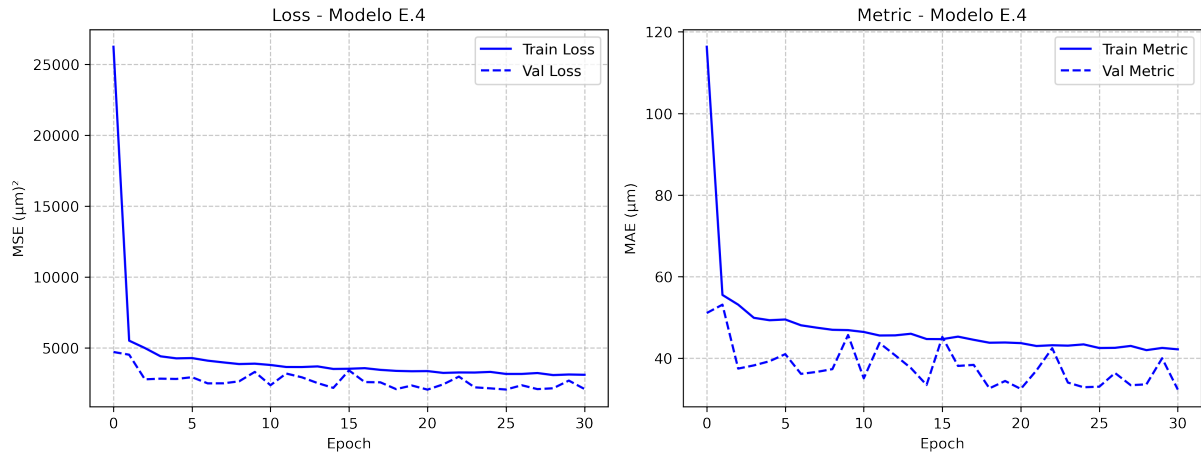


Figura 6.45 – Curvas de desempenho ao longo do treinamento e validação para o Modelo E.4.

O desempenho das curvas de treinamento e validação mostra um pequeno *underfitting* tanto para a função de perda quanto para a métrica. O modelo atingiu a convergência com 30% das *epochs*.

#### MODELO E.5

Neste modelo as taxas das camadas de *dropout* são iguais a 0.3 e 0.35. As curvas das funções de perda e métrica ao longo das *epochs* estão reportadas na figura 6.46.

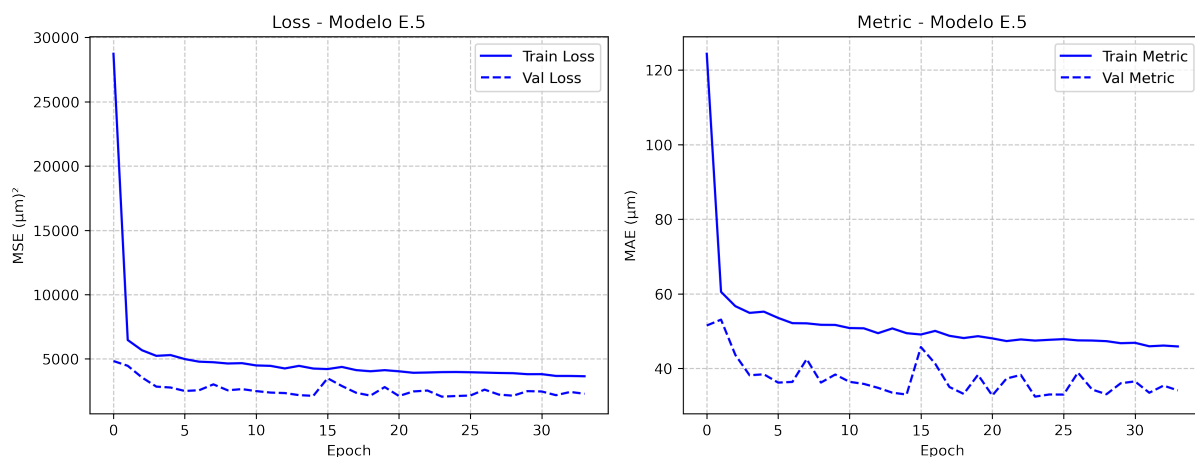


Figura 6.46 – Curvas de desempenho ao longo do treinamento e validação para o Modelo E.5.

Este modelo teve um *underfitting* mais acentuado que os demais, motivados pelas altas taxas de *dropout*. O modelo convergiu com pouco mais de 30% das *epochs*.

#### MODELO E.6

As taxas das camadas de *dropout* são iguais a 0.25 e 0.5, as maiores implementadas nesta rodada. As curvas das funções de perda e métrica estão reportadas na figura 6.47.

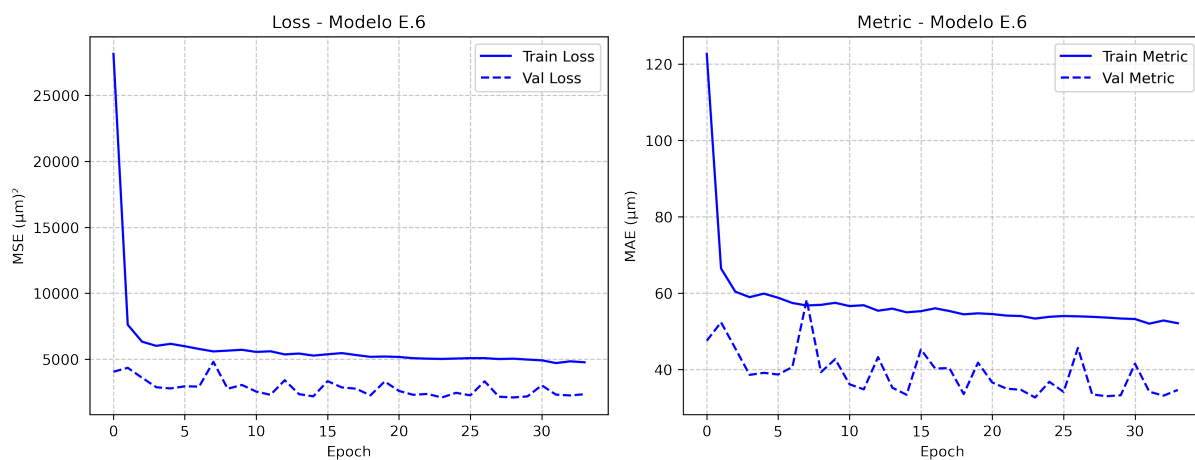


Figura 6.47 – Curvas de desempenho ao longo do treinamento e validação para o Modelo E.6.

Tanto as curvas de perda quanto as de métrica apontam para um *underfitting*, o maior observado dentro do grupo E. O modelo atingiu a convergência em pouco mais de 30% das *epochs*.

#### COMPARAÇÃO ENTRE OS MODELOS E

Os modelos desta rodada foram comparados através dos gráficos de *profile* e *residual plot*, apresentados na figura 6.48. Também usamos o de desvio padrão por *bin*, reportado

na figura 6.49. Os valores médios individuais dos modelos para a perda e métrica durante a validação estão descritos na tabela 10.

Modelo	MSE ( $\mu m$ ) <sup>2</sup>	MAE ( $\mu m$ )
E.1	2227	33.82
E.2	2086	32.39
E.3	2208	33.73
E.4	2047	32.49
E.5	2065	32.45
E.6	2102	32.65

Tabela 10 – Valores da função de perda e métrica da validação para os modelos da rodada E.

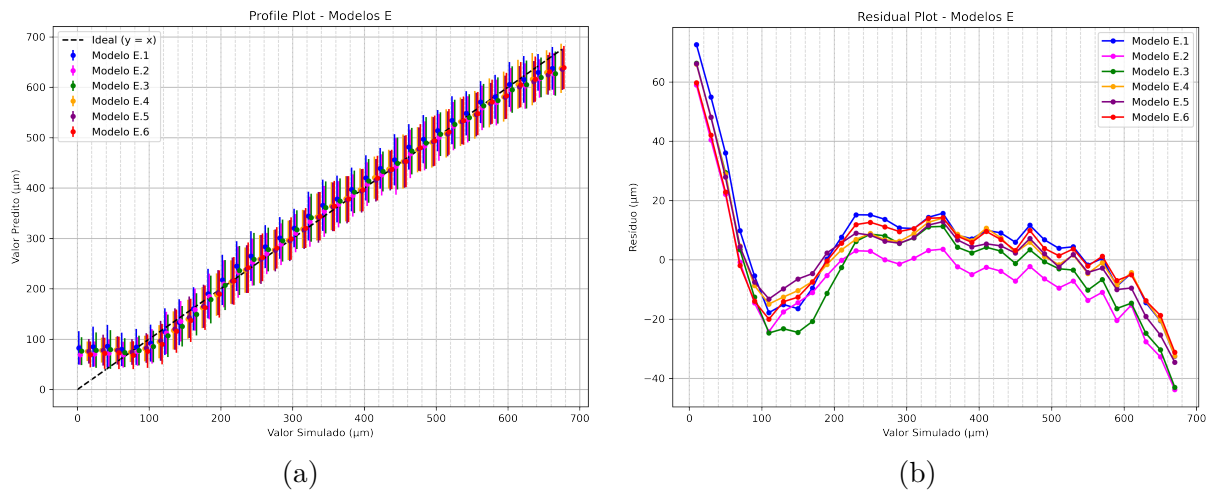


Figura 6.48 – *Profile Plot* (a) e *Residual Plot* (b) para os modelos E.

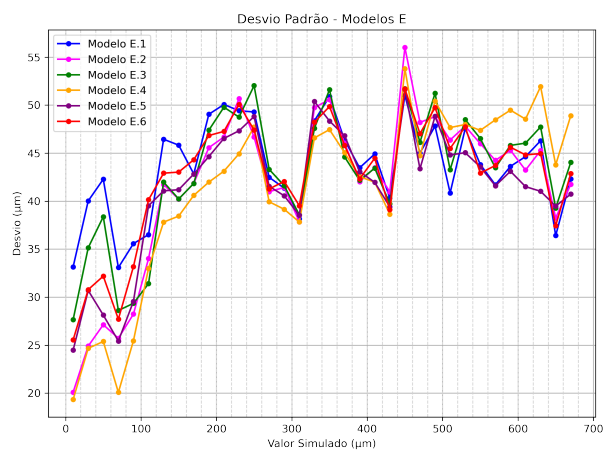


Figura 6.49 – Desvio padrão por *bin* para os modelos E.

Todos os modelos, com exceção de E.2, possuem, em média, desempenhos similares no regime de 200 a 500  $\mu m$ . Para valores maiores, os modelos E.1, E.4 e E.6 se destacam com

menores valores de resíduo. Quanto ao gráfico de desvio padrão por *bin*, o modelo E.4 tem menores dispersões na previsão de profundidades rasas. Já em regimes maiores que 500  $\mu\text{m}$ , o desvio do modelo E.4 cresce enquanto os demais decaem. Embora as curvas do modelo E.4 indiquem um *underfitting* maior que os vistos em E.1, E.2 e E.3, a sua performance na previsão de eventos com profundidades maiores que 500  $\mu\text{m}$  motivam a escolha. Além disto, o modelo E.4 possui o menor valor de MSE e está entre os que tem as menores métricas MAE.

### 6.3.7 MODELOS F

Neste grupo de modelos estudamos diferentes otimizadores e como influenciam na eficiência da predição de cada modelo. Utilizamos a arquitetura do modelo E.4 como base para esta rodada. Apenas o primeiro modelo, F.1, possui taxas de aprendizagem e *momentum* não padrão. Nos demais, estes parâmetros permaneceram como *default*.

#### MODELO F.1

Este modelo foi compilado com o otimizador *Stochastic Gradient Descent* (SGD) com *momentum* e *momentum Nesterov*. Neste modelo utilizamos uma taxa de aprendizado igual a 0.0001 e um *momentum* de 0.9. As curvas das funções de perda e métrica ao longo das *epochs* são apresentadas na figura 6.50.

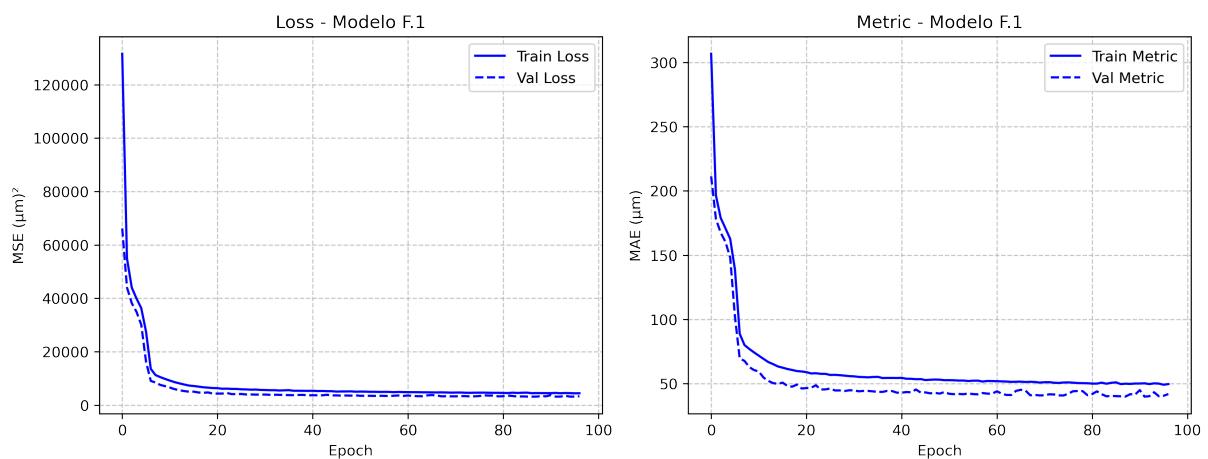


Figura 6.50 – Curvas de desempenho ao longo do treinamento e validação para o Modelo F.1.

A evolução da função de perda mostra um desempenho estável, onde as curvas de treinamento e validação coincidem. Já as funções de métrica apontam para um pequeno *underfitting*. O modelo utilizou mais de 90% das *epochs* para convergir.

## MODELO F.2

Para este modelo utilizamos o otimizador RMSprop com taxa de aprendizado padrão (igual a 0.001). As curvas das funções de perda e métrica ao longo das *epochs* são reportadas nos gráficos da figura 6.51.

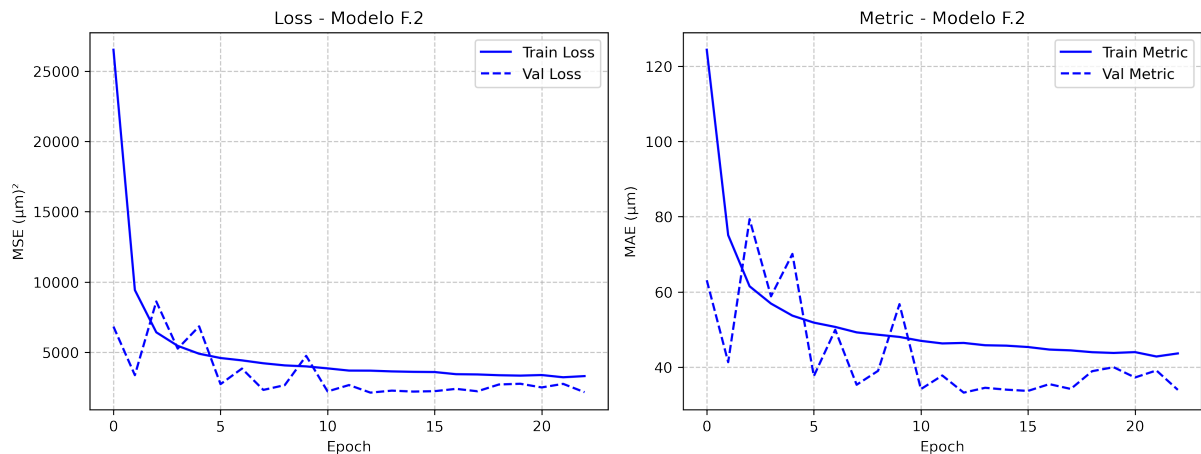


Figura 6.51 – Curvas de desempenho ao longo do treinamento e validação para o Modelo F.2.

O desempenho das curvas de treinamento e validação para os casos da perda e métrica nos mostram um pequeno *underfitting*. O modelo atingiu a convergência com pouco mais de 20% das *epochs*.

## MODELO F.3

Nesta arquitetura utilizamos o otimizador Nadam com sua taxa de aprendizado padrão (de 0.001). As curvas das funções de perda e métrica ao longo das *epochs* para o treino e validação são mostradas na figura 6.52.

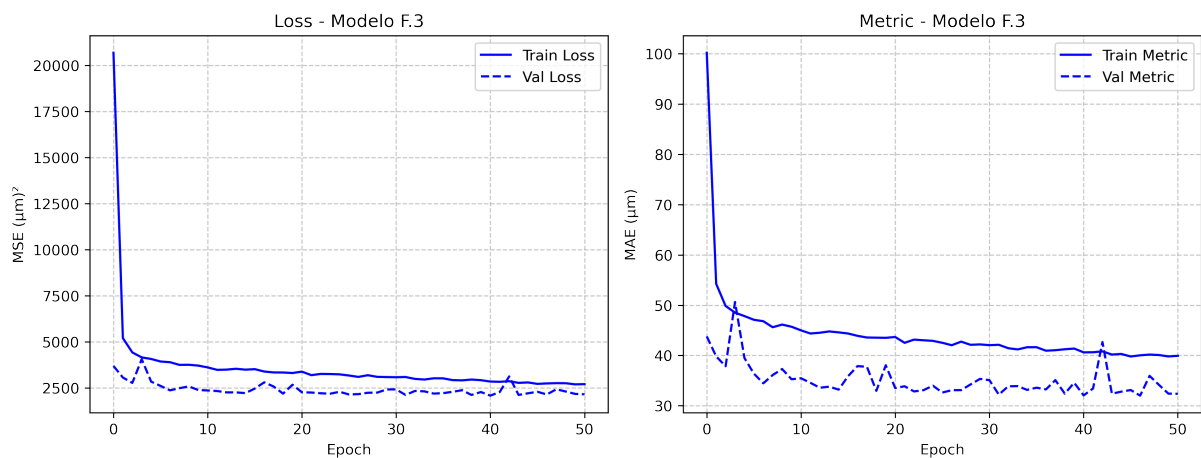


Figura 6.52 – Curvas de desempenho ao longo do treinamento e validação para o Modelo F.3.

A evolução das curvas de treinamento e validação da perda apontam para um *underfitting* pequeno, enquanto que para o caso da métrica este comportamento é mais acentuado. O modelo convergiu com 50% das *epochs* disponíveis.

#### MODELO F.4

Para este modelo utilizamos o otimizador ADAM com taxa de aprendizado padrão (0.001), o que foi implementado no treinamento de todos os modelos em rodadas anteriores a esta. As curvas de perda e métrica ao longo das *epochs* são apresentadas na figura 6.53.

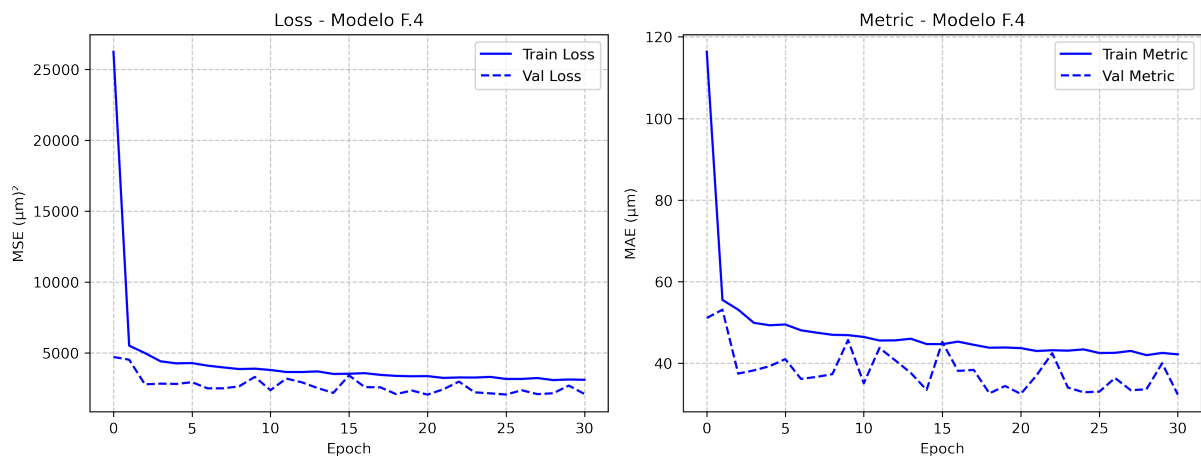


Figura 6.53 – Curvas de desempenho ao longo do treinamento e validação para o Modelo F.4.

A dinâmica das curvas de treino e validação para a perda e métrica mostram um *underfitting*. O modelo convergiu com 30% das *epochs* estipuladas.

#### COMPARAÇÃO DOS MODELOS F

A comparação destes modelos foi feita através dos gráficos de *profile* e *residual plot*, apresentados na figura 6.54. Também usamos o desvio padrão por *bin*, que está reportado na figura 6.55. Os valores médios individuais para a função de perda e métrica da validação estão descritos na tabela 11.

Modelo	MSE $(\mu m)^2$	MAE $(\mu m)$
F.1	3190	39.95
F.2	2145	33.21
F.3	2073	32.05
F.4	2047	32.49

Tabela 11 – Valores da função de perda e métrica da validação para os modelos da rodada F.

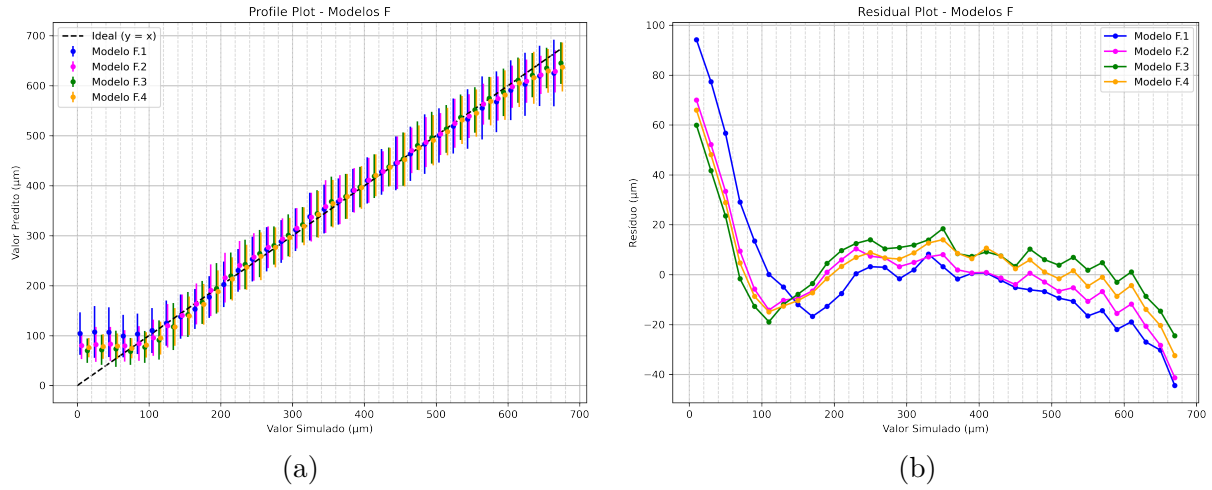


Figura 6.54 – *Profile Plot* (a) e *Residual Plot* (b) para os modelos F.

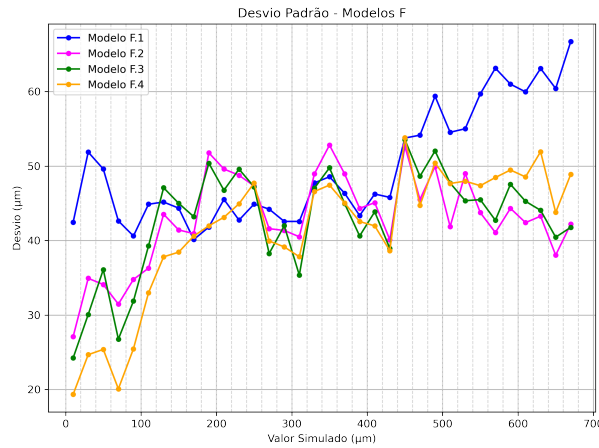


Figura 6.55 – Desvio padrão por *bin* para os Modelos F.

Analisando os gráficos de *profile* e *residual plot* fica nítido que os modelos F.1 e F.2 tem predições mais acuradas no intervalo de 200 a 500  $\mu\text{m}$ , este desempenho não se sustenta para valores maiores. Enquanto que os modelos F.3 e F.4 têm performances boas em regimes de profundidade maiores que 500  $\mu\text{m}$ , em especial o modelo F.3 se destaca positivamente neste domínio. O desvio padrão por *bin* nos mostra que os modelos F.2, F.3 e F.4 tem previsões menos dispersas no geral. O otimizador que rendeu melhor performance quanto as predições foi o Nadam, utilizado no modelo F.3. Por isso, ele foi escolhido para a rodada seguinte.

### 6.3.8 MODELOS G

Nesta última rodada nos dedicamos a estudar a influência de diferentes taxas de aprendizado (*learning rate*) combinadas com a função de otimização *Nadam*, que foi implemetada no modelo escolhido a partir da rodada anterior.



## MODELO G.1

Esta arquitetura foi implementada com o Nadam associado a uma taxa de aprendizado de 0.0001. As curvas de perda e métrica ao longo das *epochs* para o treinamento e validação estão reportadas na figura 6.56.

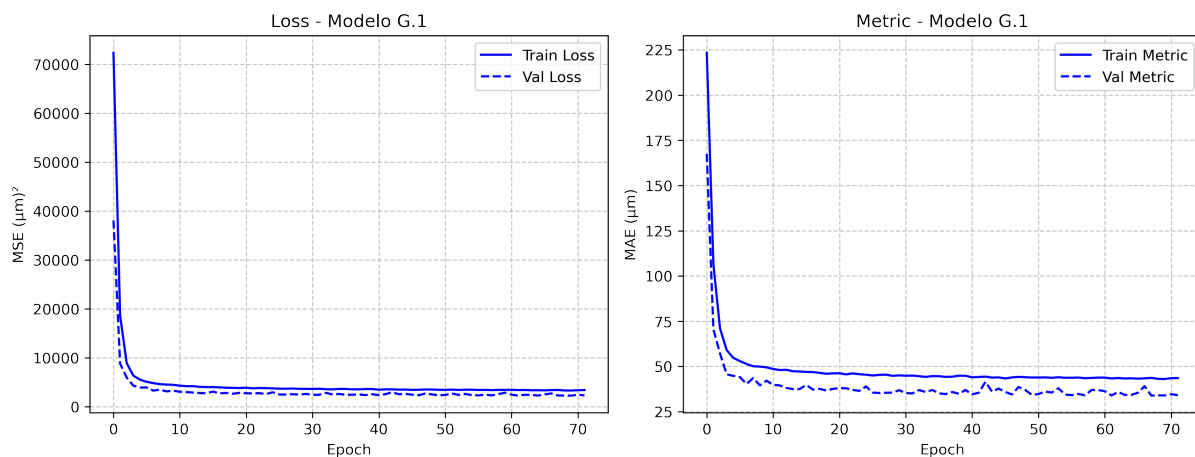


Figura 6.56 – Curvas de desempenho ao longo do treinamento e validação para o Modelo G.1.

O desempenho das curvas de perda mostram estabilidade na relação de treino e validação, que praticamente são coincidentes. No caso da métrica há um pequeno *underfitting*. O modelo utilizou pouco mais de 70% das *epochs* até convergir.

## MODELO G.2

Para este modelo o Nadam foi combinado a uma taxa de aprendizado de 0.001, que é *default* deste otimizador e foi utilizado em todos os modelos até a rodada E. As curvas das funções de perda e métrica ao longo das *epochs* são apresentados na figura 6.57.

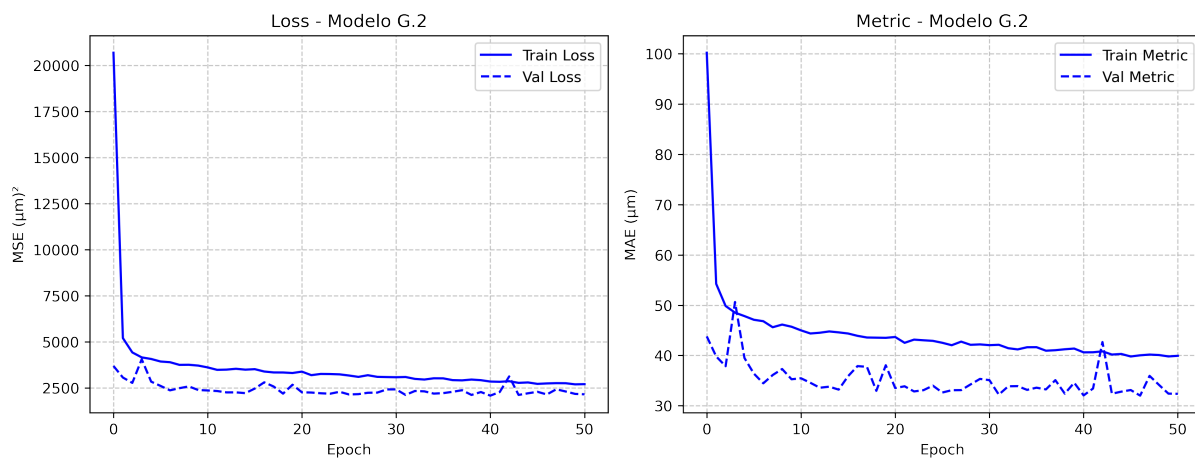


Figura 6.57 – Curvas de desempenho ao longo do treinamento e validação para o Modelo G.2.

Tantos as curvas da perda quanto as da métrica apontam para um *underfitting* do modelo, no segundo caso este comportamento é um pouco mais acentuado. A convergência foi alcançada com 50 % das *epochs* estipuladas.

### MODELO G.3

Neste modelo utilizamos o Nadam com uma taxa de aprendizado mais alta, igual a 0.01. As curvas de perda e métrica ao longo das *epochs* são reportadas na figura 6.58.

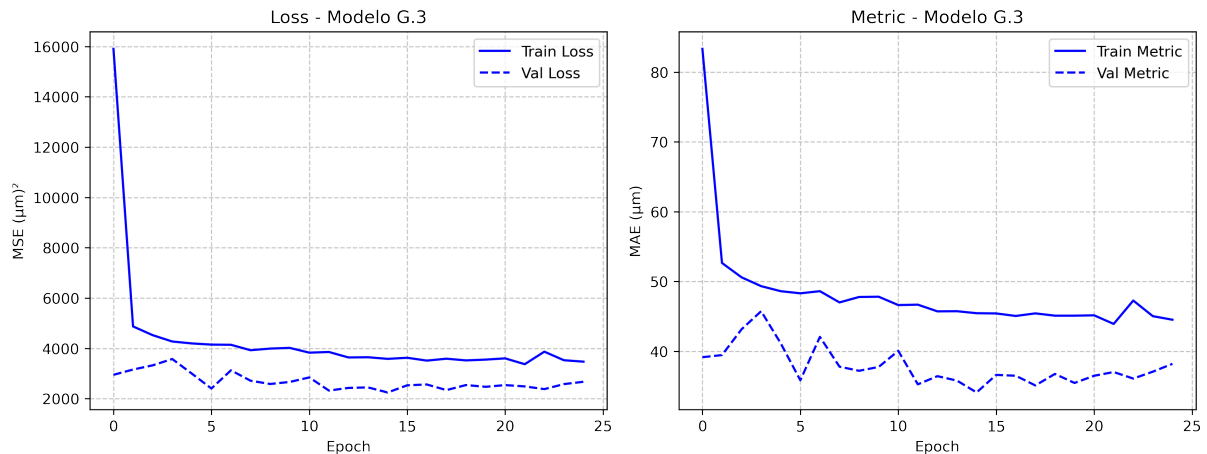


Figura 6.58 – Curvas de desempenho ao longo do treinamento e validação para o Modelo G.3.

A evolução das curvas de perda e métrica indicam um *underfitting* acentuado, o que pode ser observado tanto na função de perda quanto na métrica. O modelo convergiu com pouco menos de 25% das *epochs*.

### COMPARAÇÃO ENTRE OS MODELOS G

Para comparar os modelos utilizamos os gráficos de *profile* e *residual plot*, apresentados na figura 6.59, e o desvio padrão por *bin*, reportado na figura 6.60. A tabela 12 descreve os valores médios da perda e métrica durante a validação.

Modelo	MSE ( $\mu m^2$ )	MAE ( $\mu m$ )
G.1	2294	33.96
G.2	2073	32.05
G.3	2244	34.10

Tabela 12 – Valores da função de perda e métrica da validação para os modelos da rodada G.

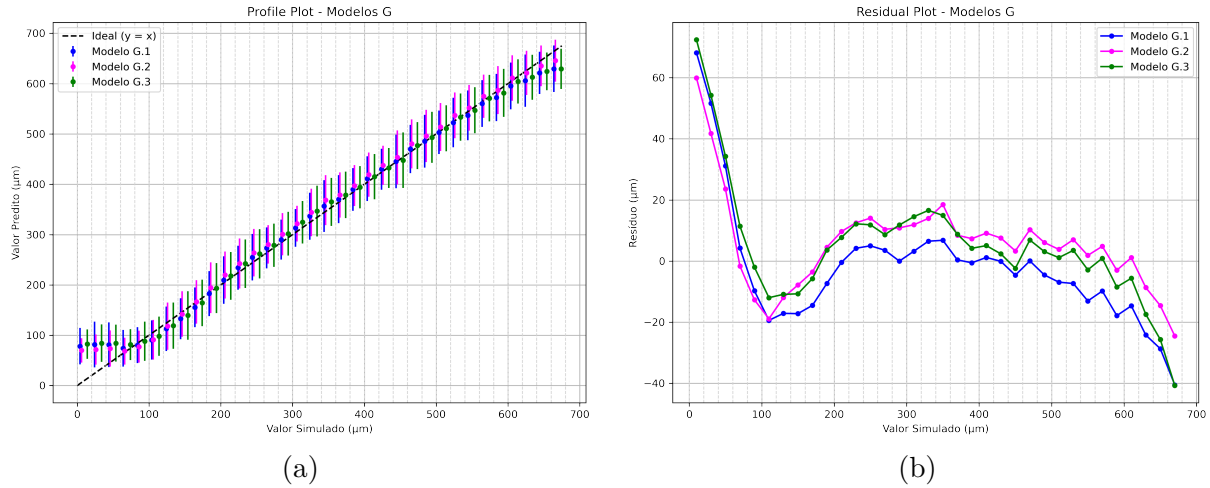


Figura 6.59 – *Profile Plot* (a) e *Residual Plot* (b) para os modelos G.

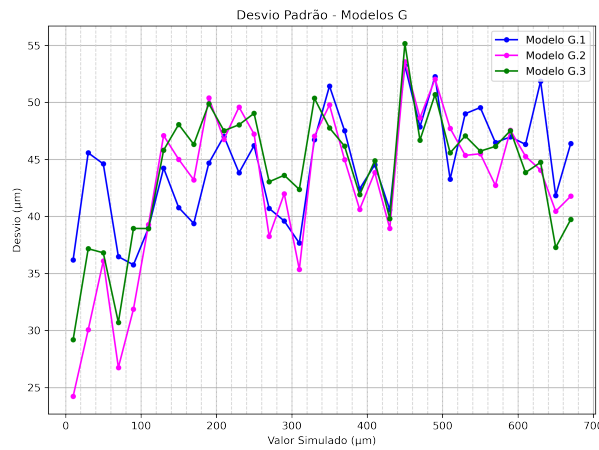


Figura 6.60 – Desvio padrão por *bin* para os Modelos G.

Os gráficos de *profile* e *residual plot* nos mostram que o modelo G.1 é o que possui predições mais acuradas no regime de 200 a 500  $\mu\text{m}$ , no entanto, esta performance não é preservada em profundidades maiores. A partir de 500  $\mu\text{m}$  o modelo G.2 possui melhor desempenho, com resíduos menores. O desvio padrão por *bin* indica que em profundidades até 100  $\mu\text{m}$  o modelo G.2 tem predições menos dispersas que os demais. No regime de 200 a 500  $\mu\text{m}$  as curvas de desvio padrão nos três modelos é muito similar. E em profundidades maiores que 500 m é o modelo G.3 que tem menor dispersão. A análise conjunta dos desempenhos individuais e dos gráficos comparativos nos motivou a escolher o modelo G.2 como a melhor arquitetura desta rodada, e, conseqüentemente como a rede mais adequada para o desafio proposto.

## 6.4 COMPARAÇÃO DO DESEMPENHO DA REDE (MODELO G.2) COM O ALGORITMO USUAL DE CONNIE

A última etapa da análise é baseada em comparar os resultados das previsões feitas com o algoritmo usual de CONNIE e com a rede neural convolucional com melhor performance, o modelo G.2, escolhido depois de um amplo estudo de variadas arquiteturas. Para comparar o algoritmo de CONNIE com a rede, fornecemos como entrada as mesmas imagens do conjunto de teste, que foi utilizado em todas as comparações anteriores. A avaliação do desempenho foi feita através dos gráficos de *profile plot* e *residual plot*, apresentados na figura 6.61. Também utilizamos o gráfico de desvio padrão por *bin*, reportado na figura 6.62.

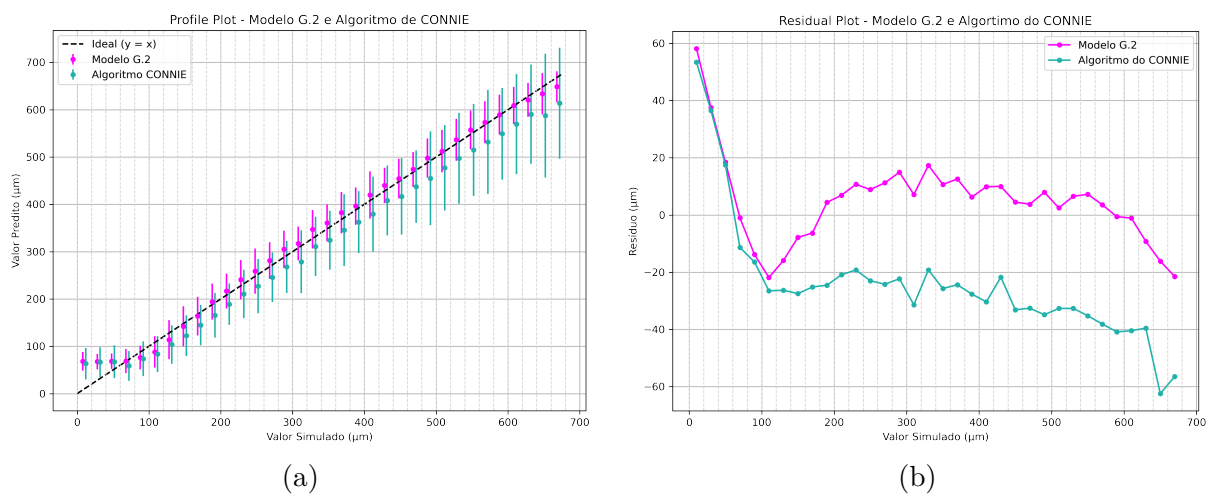


Figura 6.61 – *Profile Plot* (a) e *Residual Plot* (b) comparativos entre os resultados de predição do algoritmo de CONNIE e a rede neural.

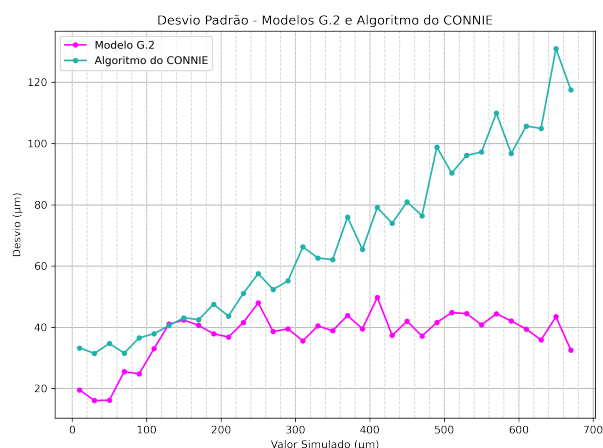


Figura 6.62 – Desvio padrão por *bin* para o algoritmo usual de CONNIE e a rede neural G.2.

Os gráficos de *profile* e *residual plot* mostram que a rede neural e o algoritmo de CONNIE possuem performances similares na predição de profundidades até pouco menos

de  $100\ \mu\text{m}$ , depois disto fica evidente que a rede neural desenvolvida possui predições mais acuradas. Enquanto que para a rede neural os resíduos oscilam em módulo em torno de  $20\ \mu\text{m}$ , para o algoritmo de CONNIE eles variam de  $-20\ \mu\text{m}$  até  $-60\ \mu\text{m}$ , com o desvio aumentando com a profundidade. O gráfico do desvio padrão por *bin* também expressa o melhor desempenho da rede neural, que é manifestado em predições menos dispersas que as do algoritmo de CONNIE em profundidades maiores que  $200\ \mu\text{m}$ . Enquanto as dispersões variam pouco com a profundidade no caso da CNN, elas aumentam sistematicamente no caso do algoritmo de CONNIE. Concluimos que, dentro das condições do presente estudo, a reconstrução da profundidade obtida a partir do melhor modelo de CNN alcançado é significativamente melhor que o algoritmo utilizado atualmente em CONNIE.

# Capítulo 7

## Considerações finais

Um dos maiores desafios atrelados a detecção de  $\text{CE}\nu\text{NS}$  está na identificação dos eventos de carga pontual, que são candidatos a esta interação. Por serem eventos de baixas energias, podem ser confundidos com sinais similares provenientes de outros tipos de fenômenos. Uma forma de caracterizar os eventos candidatos a  $\text{CE}\nu\text{NS}$  é por meio das profundidades em que foram produzidos. Atualmente, o algoritmo implementado no experimento CONNIE para fazer isso (seção 2.5) apresenta limitações nas estimativas de profundidades de eventos com poucos pixels. Neste contexto, este trabalho foi desenvolvido para superar estas fragilidades e estruturado em duas partes fundamentais: um algoritmo de simulação de imagens, utilizado para treinar e validar as redes; e a construção de redes neurais convolucionais, para previsão das profundidades dos eventos de carga pontual diretamente a partir do padrão de difusão.

Com o intuito de contribuir para a melhoria da identificação de eventos de baixas energias no experimento CONNIE, desenvolvemos e aplicamos um algoritmo de simulação de eventos pontuais sobre um plano de fundo com contribuições de ruído associadas ao pixel. Esse algoritmo foi validado através da distribuição de diferentes parâmetros, como as projeções dos baricentros e energias reconstruídas. Além disto, também comparamos o espectro real do experimento, construído a partir dos dados obtidos com o reator ligado e desligado, com o espectro simulado. O algoritmo apresenta grande utilidade em estudos da reconstrução de eventos falso positivos e na avaliação da cadeia de extração de eventos que é implementada atualmente no experimento.

Através deste mesmo algoritmo, simulamos as imagens com eventos únicos de carga pontual utilizadas para treinar e validar as redes neurais convolucionais. Promovemos um estudo exaustivo de diferentes arquiteturas de CNNs, variando hiperparâmetros distintos, para escolher a combinação que constitui a rede mais adequada para o nosso problema. Para isto, levamos em conta o desempenho do treinamento, analisando se há *overfitting* ou *underfitting*. Também consideramos os valores finais das métricas MSE e MAE para cada arquitetura desenvolvida, assim como os resíduos (viés) e as dispersões em função da profundidade. O estudo foi dividido em rodadas, na primeira investigamos o número

de camadas convolucionais e vimos que a capacidade preditiva é melhor para o modelo com maior número de camadas. Na segunda rodada, o hiperparâmetro estudado foi o número de filtros das camadas convolucionais. Neste caso, o estudo mostrou que o ideal é implementar um número intermediário de filtros, sendo que o modelo com 20 filtros por camada teve o melhor desempenho. Na terceira rodada estudamos o número adequado de neurônios na camada densa e concluímos que os modelos com maiores números de neurônios possuíam melhor performance na predição. A etapa seguinte consistiu em fazer a validação cruzada para os modelos desta terceira rodada a fim de entender como a performance de cada um dependia do conjunto de dados. Os resultados indicaram que os modelos com maior número de neurônios (100 e 200) tiveram performances uniformes em cada um dos *folds* e muito similares entre si. Para determinar o modelo mais adequado, utilizamos as mesmas arquiteturas em uma quarta rodada, que estudou a influência da adição de uma camada densa intermediária. Os resultados desta rodada mostraram que os modelos sem camada intermediária têm melhores performances. Por conta do baixo custo computacional escolhemos, por fim, o modelo com 100 neurônios para seguir com os estudos. A quinta rodada estudou a melhor combinação das taxas de dropout, os resultados apontaram que as melhores performances são decorrentes do uso de taxas moderadas, de 0.2 até 0.5, que é um intervalo comum e frequentemente utilizado. O modelo escolhido implementou duas camadas de dropout com taxas iguais a 0.2 e 0.25, respectivamente. Na sexta rodada, nos dedicamos a estudar os otimizadores. Para isto, os modelos foram treinados com SGD com *momentum*, RMSprop, ADAM e NADAM. O otimizador que rendeu melhor performance foi o NADAM. A sétima, e última rodada, estudou as taxas de aprendizagem do NADAM adequadas para o problema. Os resultados mostraram que a melhor performance foi obtida com a taxa de aprendizagem *default* deste otimizador, igual a 0.001. A partir deste estudo determinamos como melhor o modelo G.2 (seção 6.3), que conta com 6 camadas convolucionais com 20 filtros cada uma, 2 camadas de *maxpooling*, 2 camadas de *dropout* e uma camada densa com 100 neurônios. Esta arquitetura foi treinada com o otimizador NADAM associado a uma taxa de aprendizagem de 0.001. O diagrama da figura 7.1 mostra como estes blocos foram associados e a dimensionalidade de cada saída.

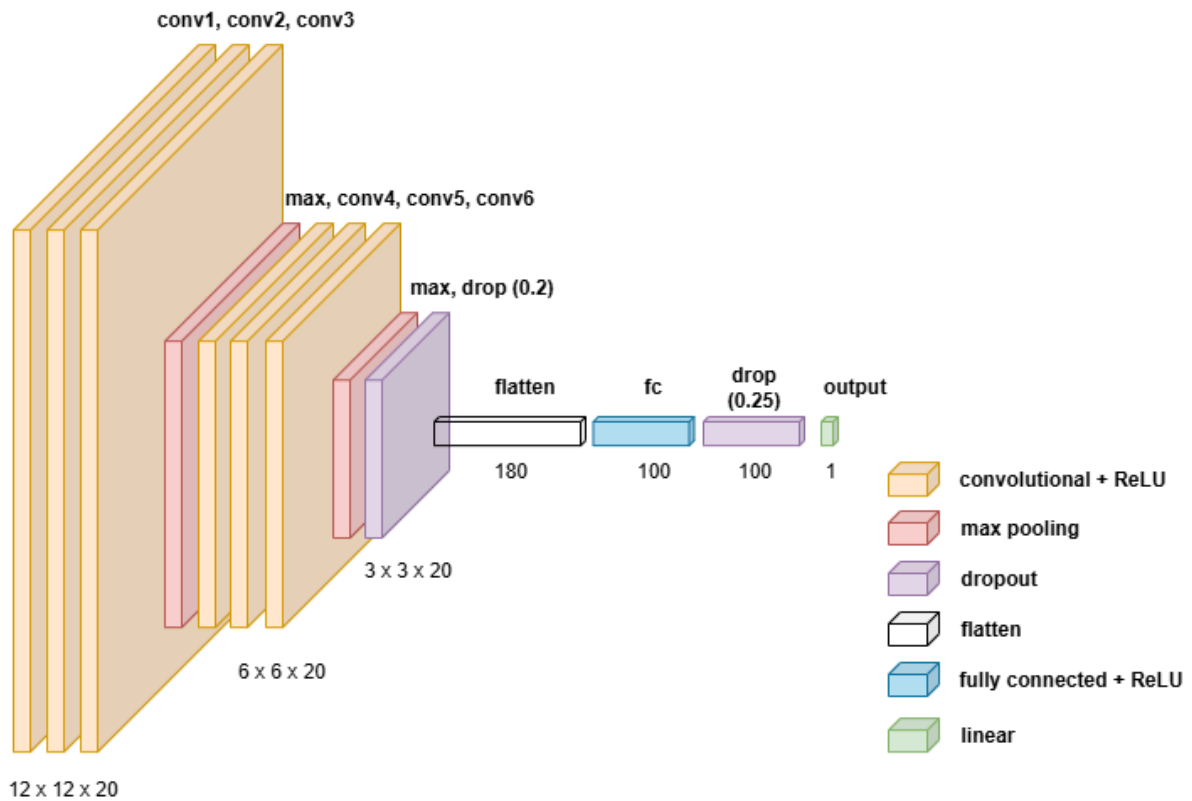


Figura 7.1 – Diagrama da arquitetura G.2, que apresentou o melhor desempenho em nossos estudos (seção 6.3).

A performance da rede neural convolucional para prever as profundidades dos eventos de carga pontual mostrou uma melhor performance que o algoritmo de CONNIE (baseado no ajuste de gaussianas). Um outro fator a ser considerado é o tempo de implementação de cada método. A CNN leva em torno de 3 minutos para ser treinada e validada, mas sua aplicação sobre o conjunto de teste (com 10 mil imagens) é extremamente rápida, demorando apenas 1.2 segundos. Enquanto isso, o algoritmo de CONNIE leva cerca de 3 minutos para processar o mesmo conjunto de imagens. Uma perspectiva futura consiste em fazer testes adicionais de comparação do desempenho para os dois métodos. Um destes testes envolve os *outliers*, definidos como predições que apresentam desvios significativos em relação ao valor esperado. Também pretendemos treinar a rede com um *dataset* simulado contendo maior proporção de eventos rasos, com profundidades de 0 a 300  $\mu\text{m}$ , para entender como isto influencia em sua performance. Vamos investigar ainda o regime de 0 a 100  $\mu\text{m}$ , onde tanto a rede neural quanto o algoritmo de CONNIE tiveram resultados imprecisos. Um último teste consistiria em adicionar todos os graus de realismo que acreditamos serem necessários, otimizar a rede e analisar os resultados finais. Se os resultados descritos neste trabalho se confirmarem, o método com CNN poderia se tornar o novo método padrão, no lugar do algoritmo de CONNIE.





# Apêndice A

## Reconstrução de Falsos Positivos

As interações candidatas ao espalhamento antineutrino-núcleo podem ser facilmente confundidas com sinais provenientes da taxa de emissão de elétron único. Por isso, é importante entender, para uma configuração específica de *clusterização*, como a taxa de emissão de elétron único afeta na classificação de eventos acima do limiar como eventos de carga pontual. Este estudo busca determinar até que nível essas taxas podem ser toleradas sem comprometer a qualidade da aquisição dos dados. Para isto, utilizamos o algoritmo de simulação desenvolvido (capítulo 4) para produzir imagens com tamanho igual a  $2000 \times 2000$  pixels, cada imagem foi simulada apenas com *background*, sem nenhuma interação de carga pontual. Os parâmetros de  $\sigma$  e  $\lambda$  foram escolhidos em torno dos valores em que o experimento CONNIE está operando, iguais a  $\sigma = 0.16 \text{ e}^-$  e  $\lambda = 0.045 \text{ e}^-/\text{px}/\text{dia}$  (que corresponde a  $0.00375 \text{ e}^-/\text{px}/\text{img}$ ). Para cada par de parâmetros foi simulada uma única imagem e para cada uma delas um catálogo do tipo root foi gerado com o número de eventos falsos reconstruídos.

Na figura A.1, temos o estudo feito para diferentes valores de ruído de leitura (num intervalo de 0.1 até  $0.3 \text{ e}^-$ ) em função dos valores de taxa de emissão de elétron único (variando de 0.010 até  $0.070 \text{ e}^-/\text{px}/\text{dia}$ ). Como podemos perceber, as imagens simuladas com os dois maiores valores de  $\sigma$  possuem maiores números de eventos falso positivos reconstruídos. O gráfico A.2 foi feito para entender com mais precisão o comportamento para os ruídos de leitura iguais a 0.10, 0.16 e  $0.20 \text{ e}^-$ . A evolução das curvas revela que o número de eventos falso positivos reconstruídos é proporcional ao valor do ruído de leitura. Este gráfico mostra ainda que nas condições de operação de CONNIE são reconstruídos cerca de 20 eventos falso positivos, o que é um valor razoável.

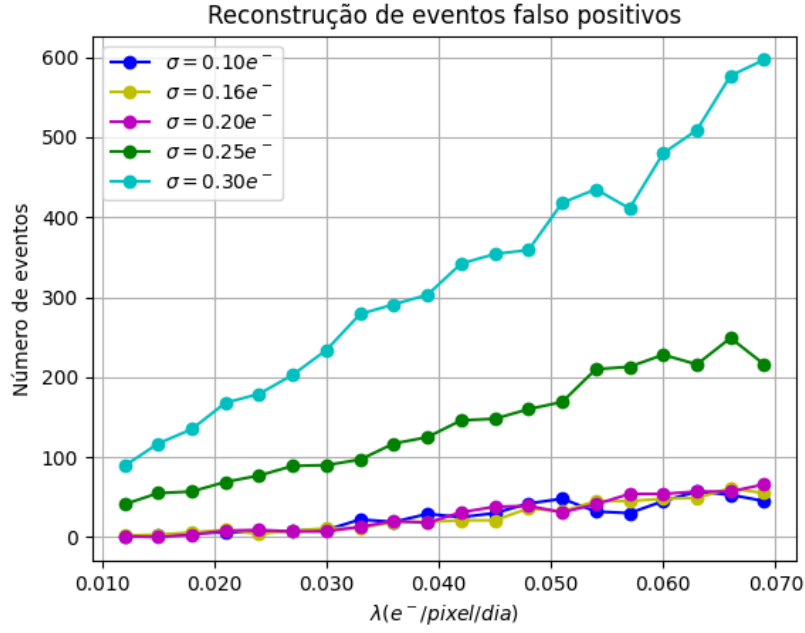


Figura A.1 – Reconstrução de eventos falso positivos para diferentes valores de ruído de leitura e taxa de emissão de elétron único.

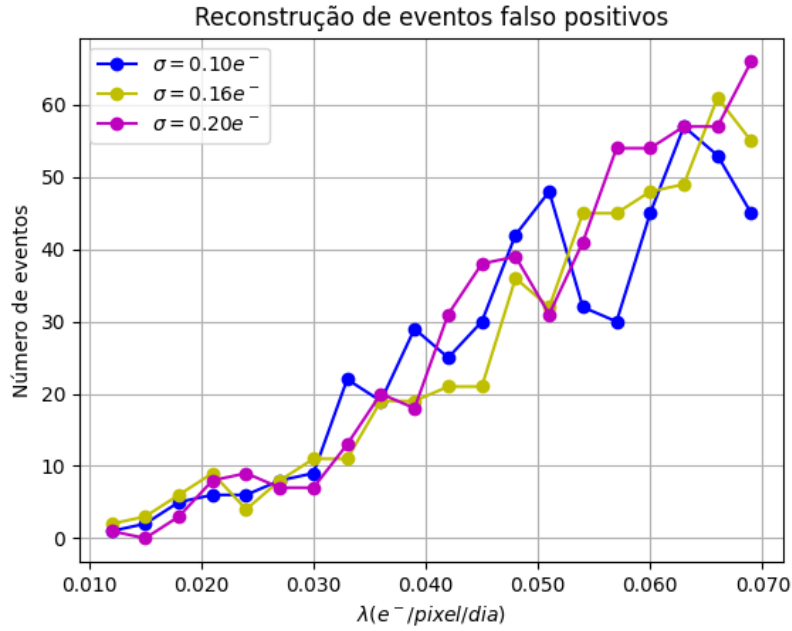


Figura A.2 – Reconstrução de eventos falso positivos focada nos valores de  $\sigma$  iguais a 0.10, 0.16 e 0.2  $e^-$ .

# Bibliografia

- [1] Laurie M. Brown. “The idea of the neutrino”. Em: Physics Today 31.9 (1978). DOI: [10.1063/1.2995181](https://doi.org/10.1063/1.2995181). URL: <https://doi.org/10.1063/1.2995181>.
- [2] Enrico Fermi. “An attempt of a theory of beta radiation. I”. Em: Zeitschrift für Physik 88 (1934), pp. 161–177. DOI: [10.1007/BF01351864](https://doi.org/10.1007/BF01351864).
- [3] John N. Bahcall, Marc H. Pinsonneault e Sarbani Basu. “Solar models: Current epoch and time dependences, neutrinos, and helioseismological properties”. Em: Astrophysical Journal 555 (2001), pp. 990–1012. DOI: [10.1086/321493](https://doi.org/10.1086/321493). arXiv: [astro-ph/0010346](https://arxiv.org/abs/astro-ph/0010346) [astro-ph].
- [4] Marco Antonio Moreira. “O Modelo Padrão da Física de Partículas (The Standard Model of Particle Physics)”. Em: Revista Brasileira de Ensino de Física 31.1 (2009), p. 1306. DOI: [10.1590/S1806-11172009000100006](https://doi.org/10.1590/S1806-11172009000100006).
- [5] C. L. Cowan et al. “Detection of the free neutrino: A Confirmation”. Em: Science 124 (1956), pp. 103–104. DOI: [10.1126/science.124.3212.103](https://doi.org/10.1126/science.124.3212.103).
- [6] G. Danby et al. “Observation of High-Energy Neutrino Reactions and the Existence of Two Kinds of Neutrinos”. Em: Phys. Rev. Lett. 9 (1962), pp. 36–44. DOI: [10.1103/PhysRevLett.9.36](https://doi.org/10.1103/PhysRevLett.9.36).
- [7] K. Kodama et al. “Observation of tau neutrino interactions”. Em: Phys. Lett. B 504 (2001), pp. 218–224. DOI: [10.1016/S0370-2693\(01\)00307-0](https://doi.org/10.1016/S0370-2693(01)00307-0). arXiv: [hep-ex/0012035](https://arxiv.org/abs/hep-ex/0012035).
- [8] R. Davis Jr., D. S. Harmer e K. C. Hoffman. “Search for Neutrinos from the Sun”. Em: Physical Review Letters 20 (1968), pp. 1205–1209. DOI: [10.1103/PhysRevLett.20.1205](https://doi.org/10.1103/PhysRevLett.20.1205).
- [9] B. Pontecorvo. “Mesonium and Antimesonium”. Em: Soviet Journal of Experimental and Theoretical Physics 6 (jan. de 1958), p. 429.

- [10] Y. Fukuda et al. “Evidence for oscillation of atmospheric neutrinos”. Em: Phys. Rev. Lett. 81 (1998), pp. 1562–1567. DOI: [10.1103/PhysRevLett.81.1562](https://doi.org/10.1103/PhysRevLett.81.1562). arXiv: [hep-ex/9807003](https://arxiv.org/abs/hep-ex/9807003).
- [11] Q. R. Ahmad et al. “Measurement of the rate of  $\nu_e + d \rightarrow p + p + e^-$  interactions produced by  $^8\text{B}$  solar neutrinos at the Sudbury Neutrino Observatory”. Em: Phys. Rev. Lett. 87 (2001), p. 071301. DOI: [10.1103/PhysRevLett.87.071301](https://doi.org/10.1103/PhysRevLett.87.071301). arXiv: [nuc1-ex/0106015](https://arxiv.org/abs/nuc1-ex/0106015).
- [12] K. et al. Eguchi. “First Results from KamLAND: Evidence for Reactor Antineutrino Disappearance”. Em: Physical Review Letters 90.2 (2003). ISSN: 1079-7114. DOI: [10.1103/physrevlett.90.021802](https://doi.org/10.1103/physrevlett.90.021802). URL: <http://dx.doi.org/10.1103/PhysRevLett.90.021802>.
- [13] Daniel Z. Freedman. “Coherent Neutrino Nucleus Scattering as a Probe of the Weak Neutral Current”. Em: Phys. Rev. D 9 (1974), pp. 1389–1392. DOI: [10.1103/PhysRevD.9.1389](https://doi.org/10.1103/PhysRevD.9.1389).
- [14] M. Cadeddu, F. Dordei e C. Giunti. “A view of Coherent Elastic Neutrino-Nucleus Scattering”. Em: arXiv (2023). v2, 26 Jul 2023. arXiv: [2307.08842](https://arxiv.org/abs/2307.08842) [[hep-ph](#)].
- [15] Manfred Lindner, Werner Rodejohann e Xun-Jie Xu. “Coherent Neutrino-Nucleus Scattering and new Neutrino Interactions”. Em: JHEP 03 (2017), p. 097. DOI: [10.1007/JHEP03\(2017\)097](https://doi.org/10.1007/JHEP03(2017)097). arXiv: [1612.04150](https://arxiv.org/abs/1612.04150) [[hep-ph](#)].
- [16] Kate Scholberg. “Coherent elastic neutrino-nucleus scattering”. Em: Journal of Physics: Conference Series. Vol. 606. 1. Abr. de 2015, p. 012010. DOI: [10.1088/1742-6596/606/1/012010](https://doi.org/10.1088/1742-6596/606/1/012010). URL: <https://dx.doi.org/10.1088/1742-6596/606/1/012010>.
- [17] Youssef Sarkis, Alexis Aguilar-Arevalo e Juan Carlos D’Olivo. “Study of the ionization efficiency for nuclear recoils in pure crystals”. Em: Phys. Rev. D 101.10 (2020), p. 102001. DOI: [10.1103/PhysRevD.101.102001](https://doi.org/10.1103/PhysRevD.101.102001). arXiv: [2001.06503](https://arxiv.org/abs/2001.06503) [[hep-ph](#)].
- [18] D. Akimov et al. “Observation of Coherent Elastic Neutrino-Nucleus Scattering”. Em: Science 357.6356 (2017), pp. 1123–1126. DOI: [10.1126/science.aao0990](https://doi.org/10.1126/science.aao0990). arXiv: [1708.01294](https://arxiv.org/abs/1708.01294) [[nuc1-ex](#)].
- [19] P.S. Barbeau, Yu. Efremenko e K. Scholberg. “COHERENT at the Spallation Neutron Source”. Em: 11 (nov. de 2021), p. 33. arXiv: [2111.07033](https://arxiv.org/abs/2111.07033) [[hep-ex](#)].
- [20] W. E. Ang, S. Prasad e S. S. Chirayath. “Antineutrino detection for temporal monitoring of fuel burnup in a large nuclear reactor”. Em:

- Nucl. Instrum. Methods Phys. Res. A 1028 (2022), p. 166353. DOI: [10.1016/j.nima.2022.166353](https://doi.org/10.1016/j.nima.2022.166353). URL: <https://doi.org/10.1016/j.nima.2022.166353>.
- [21] P. Z. M. Ventura. “Skipper-CCDs para detecção de neutrinos: primeiros resultados com o experimento CONNIE”. Master’s thesis. Universidade Federal do Rio de Janeiro, 2024, p. 70.
- [22] Alexis Aguilar-Arevalo et al. “Search for light mediators in the low-energy data of the CONNIE reactor neutrino experiment”. Em: JHEP 04 (2020), p. 054. DOI: [10.1007/JHEP04\(2020\)054](https://doi.org/10.1007/JHEP04(2020)054). arXiv: [1910.04951](https://arxiv.org/abs/1910.04951) [hep-ex].
- [23] Charles E. Chandler et al. “Sub-electron noise charge-coupled devices”. Em: Charge-Coupled Devices and Solid State Optical Sensors. Ed. por Morley M. Blouke. Vol. 1242. International Society for Optics e Photonics. SPIE, 1990, pp. 238–251. DOI: [10.1117/12.19457](https://doi.org/10.1117/12.19457). URL: <https://doi.org/10.1117/12.19457>.
- [24] A. S. Fruchter e R. N. Hook. “A novel image reconstruction method applied to deep Hubble Space Telescope images”. Em: arXiv Astrophysics e-prints (1997). eprint: [astro-ph/9708242](https://arxiv.org/abs/astro-ph/9708242). URL: <https://arxiv.org/abs/astro-ph/9708242>.
- [25] M. J. S. Belton et al. “The Galileo Solid-State Imaging experiment”. Em: Space Science Reviews 60 (1992), pp. 413–455. DOI: [10.1007/BF00216864](https://doi.org/10.1007/BF00216864). URL: <https://doi.org/10.1007/BF00216864>.
- [26] Javier Tiffenberg et al. “DAMIC: a novel dark matter experiment”. Em: 33rd International Cosmic Ray Conference. Out. de 2013, p. 1243. arXiv: [1310.6688](https://arxiv.org/abs/1310.6688) [astro-ph.IM].
- [27] Guillermo Fernandez Moroni et al. “Charge Coupled Devices for detection of coherent neutrino-nucleus scattering”. Em: Phys. Rev. D 91.7 (2015), p. 072001. DOI: [10.1103/PhysRevD.91.072001](https://doi.org/10.1103/PhysRevD.91.072001). arXiv: [1405.5761](https://arxiv.org/abs/1405.5761) [physics.ins-det].
- [28] R. D. Ryan. “Precision Measurements of the Ionization Energy and Its Temperature Variation in High Purity Silicon Radiation Detectors”. Em: IEEE Transactions on Nuclear Science 20.1 (1973), pp. 473–480. DOI: [10.1109/TNS.1973.4326950](https://doi.org/10.1109/TNS.1973.4326950).
- [29] Glenn F. Knoll. Radiation Detection and Measurement. 4th. Hoboken, NJ: Wiley, 2010. ISBN: 9780470131480. URL: <https://www.wiley.com/en-us/Radiation%2BDetection%2Band%2BMeasurement%2C%2B4th%2BEdition-p-9780470131480>.
- [30] W. D. Pence et al. “Definition of the Flexible Image Transport System (FITS), version 3.0”. Em: A&A 524, A42 (dez. de 2010), A42. DOI: [10.1051/0004-6361/201015362](https://doi.org/10.1051/0004-6361/201015362).

- [31] CERN. TFile Class Reference. 2025. URL: <https://root.cern.ch/doc/master/classTFile.html>.
- [32] Javier Tiffenberg et al. “Single-electron and single-photon sensitivity with a silicon Skipper CCD”. Em: Phys. Rev. Lett. 119.13 (2017), p. 131802. DOI: [10.1103/PhysRevLett.119.131802](https://doi.org/10.1103/PhysRevLett.119.131802). arXiv: [1706.00028](https://arxiv.org/abs/1706.00028) [[physics.ins-det](#)].
- [33] M. Sofo Haro, G. Fernandez Moroni e J. Tiffenberg. “Studies on small charge packet transport in high-resistivity fully-depleted CCDs”. Em: (2019). arXiv: [1906.11379](https://arxiv.org/abs/1906.11379) [[physics.ins-det](#)]. URL: <https://arxiv.org/abs/1906.11379>.
- [34] CONNIE Collaboration. “Exploring low-energy neutrino physics with the Coherent Neutrino Nucleus Interaction Experiment”. Em: Phys. Rev. D 100 (9 nov. de 2019), p. 092005. DOI: [10.1103/PhysRevD.100.092005](https://doi.org/10.1103/PhysRevD.100.092005). URL: <https://link.aps.org/doi/10.1103/PhysRevD.100.092005>.
- [35] A. Aguilar-Arevalo et al. “Search for low-mass WIMPs in a 0.6 kg day exposure of the DAMIC experiment at SNOLAB”. Em: Phys. Rev. D 94.8 (2016), p. 082006. DOI: [10.1103/PhysRevD.94.082006](https://doi.org/10.1103/PhysRevD.94.082006). arXiv: [1607.07410](https://arxiv.org/abs/1607.07410) [[astro-ph.CO](#)].
- [36] CONNIE Collaboration. “Results of the engineering run of the Coherent Neutrino Nucleus Interaction Experiment (CONNIE)”. Em: Journal of Instrumentation 11.07 (jul. de 2016), P07024. DOI: [10.1088/1748-0221/11/07/P07024](https://doi.org/10.1088/1748-0221/11/07/P07024). URL: <https://dx.doi.org/10.1088/1748-0221/11/07/P07024>.
- [37] Alexis A. Aguilar-Arevalo et al. “Searches for  $\text{CE}\nu\text{NS}$  and Physics beyond the Standard Model using Skipper-CCDs at CONNIE”. Em: (2024). arXiv: [2403.15976](https://arxiv.org/abs/2403.15976) [[hep-ex](#)]. URL: <https://arxiv.org/abs/2403.15976>.
- [38] LBNL Micro Systems Laboratory. <http://engineering.lbl.gov/microsystems-laboratory/>.
- [39] P. Moore et al. “MONSOON image acquisition system: control techniques for application to the orthogonal transfer array detectors”. Em: Ground-based and Airborne Instrumentation for Astronomy II. Vol. 7014. Jul. de 2008, 70147O, 70147O. DOI: [10.1117/12.802254](https://doi.org/10.1117/12.802254).
- [40] Alexis Aguilar-Arevalo et al. “Search for coherent elastic neutrino-nucleus scattering at a nuclear reactor with CONNIE 2019 data”. Em: JHEP 05 (2022), p. 017. DOI: [10.1007/JHEP05\(2022\)017](https://doi.org/10.1007/JHEP05(2022)017). arXiv: [2110.13033](https://arxiv.org/abs/2110.13033) [[hep-ex](#)].
- [41] Gustavo I. Cancelo et al. “Low Threshold Acquisition controller for Skipper CCDs”. Em: J. Astron. Telesc. Instrum. Syst. 7.1 (2021), p. 015001. DOI: [10.1117/1.JATIS.7.1.015001](https://doi.org/10.1117/1.JATIS.7.1.015001). arXiv: [2004.07599](https://arxiv.org/abs/2004.07599) [[astro-ph.IM](#)].

- [42] Irina Nasteva. “Status of the CONNIE experiment using Skipper-CCDs”. Em: (jun. de 2024).
- [43] Brenda A. Cervantes-Vergara et al. “Skipper-CCD sensors for the Oscura experiment: requirements and preliminary tests”. Em: JINST 18.08 (2023), P08016. DOI: [10.1088/1748-0221/18/08/P08016](https://doi.org/10.1088/1748-0221/18/08/P08016). arXiv: [2304.04401](https://arxiv.org/abs/2304.04401) [physics.ins-det].
- [44] Y. Sarkis, A. Aguilar-Arevalo e J. C. D’Olivo. “Ionization efficiency for nuclear recoils in silicon from about 50 eV to 3 MeV”. Em: Physical Review A 107.6 (jun. de 2023). ISSN: 2469-9934. DOI: [10.1103/physreva.107.062811](https://doi.org/10.1103/physreva.107.062811). URL: <http://dx.doi.org/10.1103/PhysRevA.107.062811>.
- [45] H. Bonet et al. “Novel constraints on neutrino physics beyond the standard model from the CONUS experiment”. Em: Journal of High Energy Physics 2022.5 (mai. de 2022). ISSN: 1029-8479. DOI: [10.1007/jhep05\(2022\)085](https://doi.org/10.1007/jhep05(2022)085). URL: [http://dx.doi.org/10.1007/JHEP05\(2022\)085](http://dx.doi.org/10.1007/JHEP05(2022)085).
- [46] M. Atzori Corona et al. “Probing light mediators and  $(g - 2)_\mu$  through detection of coherent elastic neutrino nucleus scattering at COHERENT”. Em: JHEP 05 (2022), p. 109. DOI: [10.1007/JHEP05\(2022\)109](https://doi.org/10.1007/JHEP05(2022)109). arXiv: [2202.11002](https://arxiv.org/abs/2202.11002) [hep-ph].
- [47] CONNIE collaboration. “Searches for CE $\nu$ NS and Physics beyond the Standard Model using Skipper-CCDs at CONNIE”. Em: (2024). arXiv: [2403.15976](https://arxiv.org/abs/2403.15976) [hep-ex]. URL: <https://arxiv.org/abs/2403.15976>.
- [48] Timon Emken e Chris Kouvaris. “DaMaSCUS: the impact of underground scatterings on direct detection of light dark matter”. Em: Journal of Cosmology and Astroparticle Physics 2017.10 (out. de 2017), pp. 031–031. ISSN: 1475-7516. DOI: [10.1088/1475-7516/2017/10/031](https://doi.org/10.1088/1475-7516/2017/10/031). URL: <http://dx.doi.org/10.1088/1475-7516/2017/10/031>.
- [49] CONNIE Collaboration e Atucha-II Collaboration. “Search for reactor-produced millicharged particles with Skipper-CCDs at the CONNIE and Atucha-II experiments”. Em: (2024). v1, 25 May 2024. arXiv: [2405.16316](https://arxiv.org/abs/2405.16316) [hep-ex]. URL: <https://arxiv.org/abs/2405.16316>.
- [50] Rouven Essig, Ryan Plestid e Aman Singal. “Collective excitations and low-energy ionization signatures of relativistic particles in silicon detectors”. Em: Communications Physics 7.1 (dez. de 2024). ISSN: 2399-3650. DOI: [10.1038/s42005-024-01904-2](https://doi.org/10.1038/s42005-024-01904-2). URL: <http://dx.doi.org/10.1038/s42005-024-01904-2>.
- [51] L. Singh et al. “Constraints on millicharged particles with low threshold germanium detectors at Kuo-Sheng Reactor Neutrino Laboratory”. Em: Phys. Rev. D



- 99.3 (2019), p. 032009. DOI: [10.1103/PhysRevD.99.032009](https://doi.org/10.1103/PhysRevD.99.032009). arXiv: [1808.02719](https://arxiv.org/abs/1808.02719) [hep-ph].
- [52] E. Depaoli et al. “Deployment and performance of a Low-Energy-Threshold Skipper-CCD inside a nuclear reactor”. Em: (2024). arXiv: [2401.07885](https://arxiv.org/abs/2401.07885) [hep-ex]. URL: <https://arxiv.org/abs/2401.07885>.
- [53] Michael Crisler et al. “SENSEI: First Direct-Detection Constraints on sub-GeV Dark Matter from a Surface Run”. Em: *Phys. Rev. Lett.* 121.6 (2018), p. 061803. DOI: [10.1103/PhysRevLett.121.061803](https://doi.org/10.1103/PhysRevLett.121.061803). arXiv: [1804.00088](https://arxiv.org/abs/1804.00088) [hep-ex].
- [54] Liron Barak et al. “SENSEI: Direct-Detection Results on sub-GeV Dark Matter from a New Skipper CCD”. en. Em: *Phys. Rev. Lett.* 125.17 (2020), p. 171802. ISSN: 0031-9007, 1079-7114. DOI: [10.1103/PhysRevLett.125.171802](https://doi.org/10.1103/PhysRevLett.125.171802). URL: <https://link.aps.org/doi/10.1103/PhysRevLett.125.171802>.
- [55] I. Arnquist e DAMIC-M Collaboration. “Search for Daily Modulation of MeV Dark Matter Signals with DAMIC-M”. Em: *Phys. Rev. Lett.* 132 (2024), p. 101006. DOI: [10.1103/PhysRevLett.132.101006](https://doi.org/10.1103/PhysRevLett.132.101006). arXiv: [2307.07251](https://arxiv.org/abs/2307.07251) [hep-ex].
- [56] I. Arnquist e DAMIC-M Collaboration. “First Constraints from DAMIC-M on Sub-GeV Dark-Matter Particles Interacting with Electrons”. Em: *Phys. Rev. Lett.* 130 (2023), p. 171003. DOI: [10.1103/PhysRevLett.130.171003](https://doi.org/10.1103/PhysRevLett.130.171003). arXiv: [2302.02372](https://arxiv.org/abs/2302.02372) [hep-ex].
- [57] Sebastiano Vigna. “It is high time we let go of the Mersenne Twister”. Em: *arXiv preprint arXiv:1910.06437* (2019). Submitted on 14 Oct 2019 (v1), last revised 19 Jun 2025 (this version, v3). URL: <https://arxiv.org/abs/1910.06437>.
- [58] Dario Rodrigues et al. “Absolute measurement of the Fano factor using a Skipper-CCD”. Em: *arXiv:2004.11499 [physics.ins-det]* (2020). v3, 6 Nov 2020.
- [59] Warren S. McCulloch e Walter Pitts. “A Logical Calculus of the Ideas Immanent in Nervous Activity”. Em: *The Bulletin of Mathematical Biophysics* 5 (1943), pp. 115–133. DOI: <https://doi.org/10.1007/BF02478259>.
- [60] Frank Rosenblatt. “The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain”. Em: *Psychological Review* 65.6 (1958), pp. 386–408. DOI: [10.1037/h0042519](https://doi.org/10.1037/h0042519).
- [61] Yann LeCun, Yoshua Bengio e Geoffrey Hinton. “Deep learning”. Em: *Nature* 521.7553 (2015), pp. 436–444. DOI: [10.1038/nature14539](https://doi.org/10.1038/nature14539).

- [62] Grace W. Lindsay. “Convolutional Neural Networks as a Model of the Visual System: Past, Present, and Future”. Em: Journal of Cognitive Neuroscience 33.10 (2021), pp. 2017–2031. DOI: [10.1162/jocn\\_a\\_01544](https://doi.org/10.1162/jocn_a_01544).
- [63] Mehmet Emin Asker. “Hyperspectral image classification method based on squeeze-and-excitation networks, depthwise separable convolution and multibranch feature fusion”. Em: Earth Science Informatics 16.2 (2023), pp. 1427–1448. DOI: [10.1007/s12145-023-00982-0](https://doi.org/10.1007/s12145-023-00982-0).
- [64] Shiv Ram Dubey, Satish Kumar Singh e Bidyut Baran Chaudhuri. “Activation Functions in Deep Learning: A Comprehensive Survey and Benchmark”. Em: (2022). arXiv: [2109.14545](https://arxiv.org/abs/2109.14545) [cs.LG]. URL: <https://arxiv.org/abs/2109.14545>.
- [65] Christopher M. Bishop. Pattern Recognition and Machine Learning. New York: Springer, 2006. ISBN: 978-0-387-31073-2.
- [66] Shruti Jadon Aryan Jadon Avinash Patil. “A Comprehensive Survey of Regression Based Loss Functions for Time Series Forecasting”. Em: (2022). arXiv: [2211.02989](https://arxiv.org/abs/2211.02989) [cs.LG]. URL: <https://arxiv.org/abs/2211.02989>.
- [67] David E Rumelhart, Geoffrey E Hinton e Ronald J Williams. “Learning representations by back-propagating errors”. Em: Nature 323 (1986), pp. 533–536.
- [68] Ian Goodfellow, Yoshua Bengio e Aaron Courville. Deep Learning. Cambridge, MA: MIT Press, 2016. URL: <http://www.deeplearningbook.org>.
- [69] Jun Lu. “Gradient Descent, Stochastic Optimization, and Other Tales”. Em: (2024). arXiv: [2205.00832](https://arxiv.org/abs/2205.00832) [cs.LG]. URL: <https://arxiv.org/abs/2205.00832>.
- [70] Timothy Dozat. “Incorporating Nesterov Momentum into Adam”. Em: (2016). URL: <https://openreview.net/pdf?id=OM0jvwB8jIp57ZJjtNEZ>.