Técnicas de Pré-Processamento Aplicadas no Aprendizado por Imitação para Veículos Autônomos

Tatianna Aviz[1], **Wellington Lobato**[2], **Denis Rosário**[1] **e Eduardo Cerqueira**[1]

[1] Universidade Federal do Pará (UFPA)

[2]Instituto de Computação – Universidade Estadual de Campinas (UNICAMP)

tatianna.aviz@itec.ufpa.br, wellington@lrc.ic.unicamp.br,

{denis, cerqueira}@ufpa.br

Abstract. *Autonomous driving systems often use Conditional Imitation Learning (CIL) to learn human driving behavior. In CIL, a dataset with demonstrations from a driver is used to train a model that replicates his driving behavior. However, the generalization ability of the model tends to be reduced in unfamiliar driving scenarios. Enhancing the diversity of training data via pre-processing approaches may assist in improving the capacity of the model to generalize. Therefore, this paper proposes an evaluation of data pre-processing techniques for training models based on CIL. The results demonstrate that, together, data augmentation and normalization techniques increase the generalization capacity of CIL, obtaining a 62.81% better value compared to other approaches.*

Resumo. *Os sistemas de condução autônoma utilizam frequentemente o Aprendizado por Imitação Condicional (Conditional Imitation Learning (CIL)) para aprender o comportamento da direção humana. No CIL, um conjunto de dados com demonstrações de um condutor é usado para treinar um modelo que replica o seu comportamento de condução. No entanto, a capacidade de generalização do modelo tende a ser reduzida em cenários de condução desconhecidos. Aumentar a variabilidade dos dados de treinamento por meio de técnicas de pré-processamento pode auxiliar na generalização do modelo. Portanto, neste artigo é proposto uma avaliação de técnicas de pré-processamento de dados no treinamento de modelos baseados no CIL. Os resultados demonstram que, em conjunto, as técnicas de aumentação de dados e normalização aumentam a capacidade de generalização do CIL, obtendo um valor 62.81% melhor em comparação outras abordagens.*

1. Introdução

O mercado de Veículos Autônomos (*Autonomous Vehicles (AVs)*) está previsto para alcançar a marca de 77 bilhões de dólares até 2035, com algumas estimativas sugerindo que esse valor pode até mesmo ultrapassar os 200 bilhões de dólares [Le Mero et al. 2022]. Os benefícios dos AVs vão além do conforto, incluindo melhorias significativas na segurança rodoviária, eficiência do tráfego e utilização otimizada de recursos públicos, como estradas e transporte coletivo [Eraqi et al. 2022]. Esses benefícios são cruciais para lidar com os desafios crescentes relacionados ao transporte, como congestionamento urbano, poluição do ar e melhoria da eficiência geral da mobilidade [Eraqi et al. 2022].

Para alcançar uma transição para um ambiente de condução autônoma, a pesquisa em AVs explora dois principais paradigmas: modular e ponta a ponta [Le Mero et al. 2022]. O paradigma modular segmenta o sistema dos AVs em módulos, cada um responsável por uma sub-tarefa específica de condução. Esta abordagem oferece um alto nível de verificabilidade, pois a saída de cada módulo pode ser avaliada individualmente [Zhang et al. 2020]. Em contraste, o paradigma ponta a ponta adota uma abordagem mais integrada, a percepção e o controle são aprendidos simultaneamente usando uma rede neural profunda. Dessa forma, concebendo o AV como um sistema holístico no qual todas as funcionalidades estão interligadas e otimizadas em conjunto. Essa abordagem é auto-otimizável e aprende a ser computacionalmente eficiente [Xiao et al. 2022].

O Aprendizado por Imitação (*Imitation Learning (IL)*) é uma técnica de Aprendizado de Máquina (*Machine Learning (ML)*) frequentemente aplicada em sistemas que seguem o paradigma ponta a ponta. O IL utiliza exemplos de condução fornecidos por condutores para treinar um modelo, que traduz as entradas sensoriais dos AVs em sinais de controle de baixo nível, como aceleração e ângulo de direção [Ly and Akhloufi 2021]. O IL pode aprender diretamente com a abundância de veículos movidos por humanos, sem exigir dados extras rotulados manualmente. Além disso, sistemas de IL podem ser aprendidos *off-line* de maneira segura, em contraste com abordagens de aprendizagem por reforço que normalmente exigem milhões de execuções de tentativa e erro no ambiente alvo ou uma simulação [Zheng et al. 2022].

O IL assume que o modelo treinado é capaz de mapear qualquer entrada sensorial diretamente em um sinal de controle. No entanto, existem cenários de condução em que uma única entrada sensorial pode resultar em diversos sinais de controle. Neste caso, o modelo não tem conhecimento suficiente para predizer a ação correta e escolhe arbitrariamente o próximo movimento. Alguns trabalhos propuseram adicionar um comando de entrada ao IL, esta abordagem foi denominada *Conditional Imitation Learning (CIL)*, que durante o treinamento direciona ao modelo a tomada de decisão do condutor e durante o teste pode ser usado para auxiliar na predição do modelo [Codevilla et al. 2018b, Codevilla et al. 2019].

Embora o CIL demonstre alto desempenho em cenários de condução semelhantes aos exemplos fornecidos durante o treinamento, os modelos tendem a falhar na generalização para cenários não observados anteriormente [Hawke et al. 2020]. Portanto, para garantir a capacidade de generalização, é essencial fornecer ao modelo conjuntos de dados mais diversos, não restringindo o desempenho do sistema [Ly and Akhloufi 2021]. Nesse contexto, este artigo apresenta um estudo comparativo das principais abordagens de pré-processamento de dados: normalização e aumentação de dados. As avaliações consideram as principais técnicas de aumentação de dados, tais como: Desfoque Gaussiano, Ruído Gaussiano, *Random Erasing*, Brilho e Contraste [Maharana et al. 2022]. Foram realizados experimentos com o simulador Car Learning to Act (CARLA) com o objetivo de quantificar o desempenho das técnicas de pré-processamento aplicadas. Os resultados mostram que o modelo CIL com aumentação e normalização de dados obteve resultados superiores em relação aos demais modelos avaliados, com reduções de até 62.81% de Mean Squared Error (MSE).

O restante deste artigo está organizado conforme descrito a seguir. A Seção 2 apresenta os principais trabalhos relacionados a este artigo. A Seção 3 introduz o funci-

onamento do modelo de CIL e as técnicas de pré-processamento de dados utilizadas para o treinamento dos modelos. A Seção 4 descreve o cenário de treinamento e avaliação dos modelos e discute os resultados obtidos. Por fim, a Seção 5 conclui o artigo e aponta as principais direções para os trabalhos futuros.

2. Trabalhos Relacionados

Codevilla et al. propuseram o CIL com o objetivo de condicionar o IL à entrada de comandos de alto nível [Codevilla et al. 2018b, Codevilla et al. 2019]. De acordo com a técnica proposta pelos autores, o modelo de condução recebe não apenas a entrada perceptiva e o sinal de controle, mas também uma representação da intenção do condutor. Durante o treinamento, os comandos resolvem ambiguidades e facilitam o aprendizado. Na etapa de teste, os comandos servem como canal de comunicação que pode ser utilizado para direcionar o modelo.

Eraqi et al. introduziram um modelo que aprimora o CIL, mesclando uma entrada de sensor LIDAR com dados de imagem coletados por uma câmera. O objetivo do modelo é aumentar a capacidade de generalização dos AVs para novos ambientes e para diferentes condições climáticas, problemáticas enfrentadas pelo CIL [Eraqi et al. 2020]. Os autores também apresentaram um método de mapeamento em grades de ocupação usado para retificar a saída do modelo. O sistema proposto é avaliado no simulador CARLA e demonstra melhorias na taxa de sucesso de condução autônoma e na distância média percorrida até o destino.

Hu et al. apresentaram uma abordagem de IL, chamada MILE, para aprender em conjunto um modelo do mundo e uma modelo de condução autônoma através da geometria 3D e vídeos de alta resolução das demonstrações do condutor [Hu et al. 2022]. O modelo prevê estados e ações diversos, que podem ser interpretados por meio de segmentação semântica panorâmica. Além disso, MILE pôde executar manobras de condução complexas a partir de planos inteiramente previstos. MILE demonstrou um desempenho superior em comparação com os demais métodos na pontuação de condução no simulador CARLA, mesmo em novos cenários e diferentes condições climáticas.

Teng et al. desenvolveram um modelo de dois-estágio chamado HIIL para solucionar o problema da estabilidade e interpretabilidade do AVs em cenários de condução urbana [Teng et al. 2023]. No primeiro estágio, um modelo BEV (*Bird's Eye View*) pré-treinado é usado para prover uma interpretação do ambiente ao redor do veículo usando uma máscara BEV. No segundo estágio, um modelo IIL (*Interpretable Imitation Learning*) é construído para mesclar o BEV a um ângulo de direção gerado pelo algoritmo *Pure-Pursuit*. Como comprovado através de experimentos no simulador CARLA, HIIL demonstrou interpretabilidade, generalização e robustez em cenários de navegação desconhecidos.

A partir da análise do estado da arte, identificou-se a importância da capacidade de generalização dos modelos de IL e CIL, uma vez que para tarefas críticas de segurança, como a condução autônoma, casos extremos e cenários desconhecidos representam ameaças significativas ao desempenho. Muitos dos trabalhos existentes [Eraqi et al. 2020, Hu et al. 2022, Teng et al. 2023] buscam aprimorar a performance de generalização através da arquitetura do modelo em si, sem levar em consideração o conjunto de treino. Além disso, algumas abordagens [Codevilla et al. 2018b, Codevilla et al. 2019] não ava-

liam o impacto da normalização e aumentação de dados na capacidade de generalização do modelo.

3. Aprendizado por Imitação Condicional com Aumentação de Dados

Esta Seção descreve o funcionamento das técnicas IL e CIL, bem como a arquitetura dos modelos avaliados. Além disso, são definidas as técnicas de pré-processamento de dados.

3.1. Modelagem do Sistema

O IL é uma técnica de ML que utiliza demonstrações de condução de um condutor para treinar um modelo que replica o seu comportamento. Dessa forma, no IL as demonstrações do condutor são dadas por pares observação-ação, onde é assumido que existe uma função E que mapeia cada observação o em uma ação a, tal que $a = E(o)$. No entanto, em alguns cenários de condução, uma observação pode ser mapeada em diferentes ações. Um exemplo disso ocorre quando o AV se aproxima de uma interseção: a ação subsequente não depende apenas da observação, mas é também afetada pelo estado interno ou latente do motorista. Sem ter conhecimento sobre esse estado, o AV poderia tomar decisões arbitrárias ao chegar em interseções.

O estado latente pode ser modelado por um vetor h, que, junto a observação, explica a ação do condutor: $a = E(o, h)$. O vetor h é adicionado no CIL através de um comando de entrada $c = c(h)$, que, embora não descreva todo o estado latente h, provê informações sobre a tomada de decisão do condutor. No CIL, o modelo interage com o ambiente em instantes de tempo discretos, tal que para cada instante t, ele recebe uma observação o_t e um comando de entrada c_t e retorna uma ação a_t que afeta o ambiente e gera a próxima observação. Essa configuração é ilustrada na Figura 1. No contexto da condução autônoma, o é uma entrada sensorial do AV e a um sinal de controle de baixo nível, como aceleração e ângulo de direção do volante. O comando c fornece durante o treinamento informações sobre a tomada de decisão do condutor e durante o teste pode ser usado para auxiliar a predição do modelo.

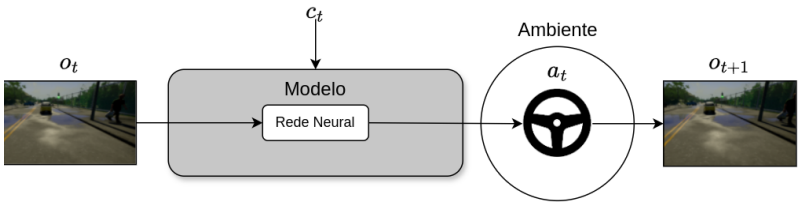


Figura 1. Visão geral do funcionamento do CIL

O *dataset* de treino do CIL gerado pelo condutor é definido como $\mathcal{D} = \{\langle o_i, c_i, a_i \rangle\}_{i=1}^{N}$, no qual N é o total de demonstrações. O treinamento é realizado mediante aprendizado supervisionado e consiste em ajustar os parâmetros θ de uma função do modelo $F(o_i, c_i; \theta)$ de modo a otimizar o mapeamento de observações e comandos em ações. O objetivo do CIL é matematicamente expresso na Equação 1, onde ℓ é a função de perda calculada entre a predição do modelo e a ação real.

$$\underset{\theta}{\text{minimize}} \sum_{i} \ell(F(o_i, c_i; \theta), a_i). \tag{1}$$

3.2. Arquitetura do IL

A rede empregada no IL é baseada no modelo proposto em [Codevilla et al. 2018b], sendo composta por quatro módulos de rede: um módulo de percepção focado no processamento das entradas de imagem, um módulo de medição que processa a entrada de velocidade, um módulo de junção que mescla as informações de percepção e medição e um módulo de controle que produz comandos motores a partir da saída do módulo de junção. Algumas das camadas que compõem os módulos são recorrentes ao longo da arquitetura, portanto foram agrupadas em dois blocos: o bloco de convolução e o bloco de camadas totalmente conectadas. Em ordem topológica, o bloco de convolução contém as seguintes camadas:

1. **Camada de convolução**: a dimensão do *kernel* e o seu *stride* variam ao longo dos blocos de convolução. Após a convolução, a função de ativação Rectified Linear Unit (ReLU) é aplicada.
2. **Camada de *max pooling***: a janela de *pooling* tem dimensão de 1×1 e o *stride* do *poling* é de 1×1.
3. **Camada de normalização**: efetua a normalização em lote.
4. **Camada de *Dropout***: probabilidade de *dropout* de 20% dos neurônios das camadas anteriores.

O bloco de camadas totalmente conectadas é composto por duas camadas:

1. **Camada densa**: após a camada densa a função de ativação ReLU é aplicada.
2. **Camada de *Dropout***: probabilidade de *dropout* de 50% dos neurônios da camada anterior.

A entrada da rede é um par de dados contendo uma imagem RGB de 200×88 *pixels*, i, e a medição da velocidade atual do AV, m. Cada entrada é processada por um módulo de rede independente, havendo um módulo de imagem I(i) e um módulo de medição M(m). I(i) é uma Rede Neural Convolucional, composta por oito blocos de convolução seguidos por uma camada *Flatten* e dois blocos de camadas totalmente conectadas. Por sua vez, M(m) é constituído somente por dois blocos de camadas totalmente conectadas. O módulo de junção J(i, m) é adicionada para unir as saídas de I(i) e M(m) e é formado por uma camada de concatenação e um bloco de camadas totalmente conectadas. A saída de J(i, m) é processada pelo módulo de controle, composto por dois blocos de camadas totalmente conectadas, e então submetida a uma camada Densa que resulta no valor de saída da rede, correspondente ao ângulo de direção.

3.3. Arquitetura do CIL

A arquitetura utilizada no CIL é semelhante ao modelo de IL sendo composta pelos mesmos módulos de percepção, medição e junção. No entanto, o módulo de controle é subdividido em quatro ramificações idênticas (*branches*). As *branches* são módulos de rede especializados em comportamentos de condução preestabelecidos. Cada *branch* é formada por dois blocos de camadas totalmente conectadas seguido por uma camada Densa que resulta no ângulo de direção. Para selecionar qual *branch* será utilizada, um conjunto discreto de comandos de alto nível $C = \{c^0, ..., c^k\}$ é introduzido, onde k é o total de *branches*.

Os modelos baseados no CIL treinam com um subconjunto dos dados, contendo somente os dados referentes a uma *branch* específica. São considerados quatro comandos

de alto nível (C): *Follow* para seguir a direção da estrada; *Left* para dobrar à esquerda na próxima interseção; *Right* para dobrar à direita na próxima interseção e *Straight* para seguir em linha reta na próxima interseção.

3.4. Modelos Avaliados

Foram desenvolvidos quatro modelos distintos com base nas arquiteturas propostas. O modelo de referência, *baseline*, foi treinado através do IL, com os dados de entrada normalizados. Este modelo emprega a arquitetura base do IL e utiliza todo o conjunto de dados durante o treino, pois não é preciso subdividir os dados entre *branches* como no CIL. Os demais modelos, por sua vez, baseiam-se na arquitetura do CIL. Eles são categorizados da seguinte forma: o modelo 1 é caracterizado pela ausência de pré-processamento dos dados; o modelo 2 aplica a técnica de normalização dos dados de entrada; e o modelo 3 inclui tanto a normalização quanto a aumentação dos dados.

A normalização de dados é aplicada tanto nos *pixels* das imagens quanto na velocidade. Esses dados de entrada, tem seus valores ajustados para o intervalo [0,1]. A Equação 2 expressa a normalização dos dados, tal que X_{norm} é o dado de entrada normalizado, X corresponde ao valor original da entrada e max_value é o valor máximo da entrada.

$$X_{norm} = \frac{X}{max\_value} \tag{2}$$

Além disso, foram aplicadas técnicas de aumentação de dados em 25% das imagens enviadas para a rede, resultando na criação de 3.750.000 imagens adicionais com diversas transformações de diferentes intensidades. Cada transformação tem uma probabilidade de 20% de ser aplicada de forma independente em cada canal de cor. Ademais, as transformações são cumulativas e ocorrem em ordem aleatória. A Figura 2 exemplifica as transformações aplicadas a uma imagem do conjunto de dados.

Figura 2. Transformações de *random erasing* e contraste aplicadas a uma imagem do conjunto de dados

A seguir são descritas todas as transformações aplicadas as imagens:

- **Desfoque Gaussiano**: probabilidade de 9% desfoque, cuja intensidade varia aleatoriamente entre 0 a 1,5. O desfoque é o resultado da convolução entre a imagem e um *kernel* gaussiano bidimensional, expresso pela seguinte equação:

$$G(x, y) = \frac{1}{\sigma\sqrt{2\pi}} \times e^{-\frac{(x^2+y^2)^2}{2\sigma^2}} \tag{3}$$

  Onde $G(x, y)$ é o valor do *kernel* na posição (x ,y) e σ é o desvio padrão que determina a amplitude do desfoque próximo a uma área.

- **Ruído Gaussiano**: probabilidade de adição de ruído de 9%. A escala do ruído varia aleatoriamente entre 0,0 e 0,05. O ruído gaussiano é calculado pela seguinte função de densidade de probabilidade:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \times e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (4)$$

  Onde x é uma variável aleatória, μ é a média da distribuição gaussiana e σ o desvio padrão.
- ***Random Erasing* (por pixel)**: probabilidade de *dropout* dos *pixels* de 30%. Apaga aleatoriamente entre 0% e 10% dos *pixels*.
- ***Random Erasing* (por região)**: probabilidade de *dropout* de regiões da imagem de 30%. Apaga aleatoriamente entre 0% a 10% dos *pixels*, em áreas cujo tamanho varia de 8% a 20% do tamanho total da imagem.
- **Brilho (por multiplicação)**: probabilidade de alteração do brilho de 40%. É realizada a multiplicação dos *pixels* de uma imagem por um valor aleatório entre 0.1 e 2.5.
- **Brilho (por adição)**: probabilidade de alteração do brilho de 30%. Adiciona um valor aleatório entre -40 e 40 aos *pixels* da imagem.
- **Contraste (linear)**: probabilidade de alteração no contraste de 9%. É aplicado um ajuste linear de contraste às imagens, em que fator de contraste é aleatoriamente selecionado entre 0.5 e 1.5.

4. Avaliação

Esta Seção apresenta a avaliação dos modelos de IL e CIL com as técnicas de pré-processamento. Além disso, são apresentados a descrição do cenário de avaliação, as configurações utilizadas e os resultados obtidos em termos de MSE, Root Mean Squared Error (RMSE) e Mean Absolute Error (MAE).

4.1. Descrição do Cenário de Avaliação

O simulador CARLA [1] foi utilizado na aquisição do conjunto de dados destinado ao treinamento e teste dos modelos. O CARLA é um simulador de código aberto desenvolvido para pesquisa em condução autônoma e sistemas de assistência ao motorista. Este simulador proporciona um ambiente de simulação realista e adaptável que viabiliza o desenvolvimento e a avaliação de algoritmos de controle para AVs [Dosovitskiy et al. 2017]. O conjunto de dados foi gerado por meio da simulação de um veículo que executa uma trajetória de condução em diferentes horários do dia e condições climáticas. Durante o percurso, foram registradas as observações sensoriais do veículo, provenientes de câmeras e outros dispositivos sensoriais.

No total, foram adquiridas aproximadamente 14 horas de dados de condução, nos quais, em 80% dos dados o AV é controlado por um agente automatizado e em 20% por um condutor humano. O conjunto de dados é composto por imagens e medições dos sensores. As imagens tem a resolução de 200×88 *pixels* e retratam a percepção do veículo, algumas destas imagens podem ser visualizadas na Figura 3. Adicionalmente, foram coletados 28 tipos de medições provenientes dos sensores, que representam o estado do veículo, incluindo, ângulo de direção, velocidade e aceleração.

[1] `http://carla.org/`

Figura 3. Amostra de imagens do conjunto de dados

O ângulo de direção é uma medida que indica quanto o volante está rotacionado em relação à sua posição central. No conjunto de dados, essa medida é dada por um valor real no intervalo [-1.0,1.0], onde valores extremos indicam rotação total para esquerda e para direita, respectivamente. Outra entrada contida no conjunto de dados é o comando de alto nível que expressa a intenção de condução do motorista em um *data point*. Os modelos avaliados foram submetidos a um treinamento de 250 épocas, com 500 *steps* por época, adotando-se um *batch* de 120, taxa de aprendizado de 2×10^{-4} e otimizador Adaptive Moment Estimation (Adam). O cenário para a realização das avaliações foi configurado utilizando um processador *Intel Core i7-8700k* com frequência de 5.10 GHz, memória RAM de 16 GiB e placa de vídeo *NVIDIA GeForce RTX 3070*.

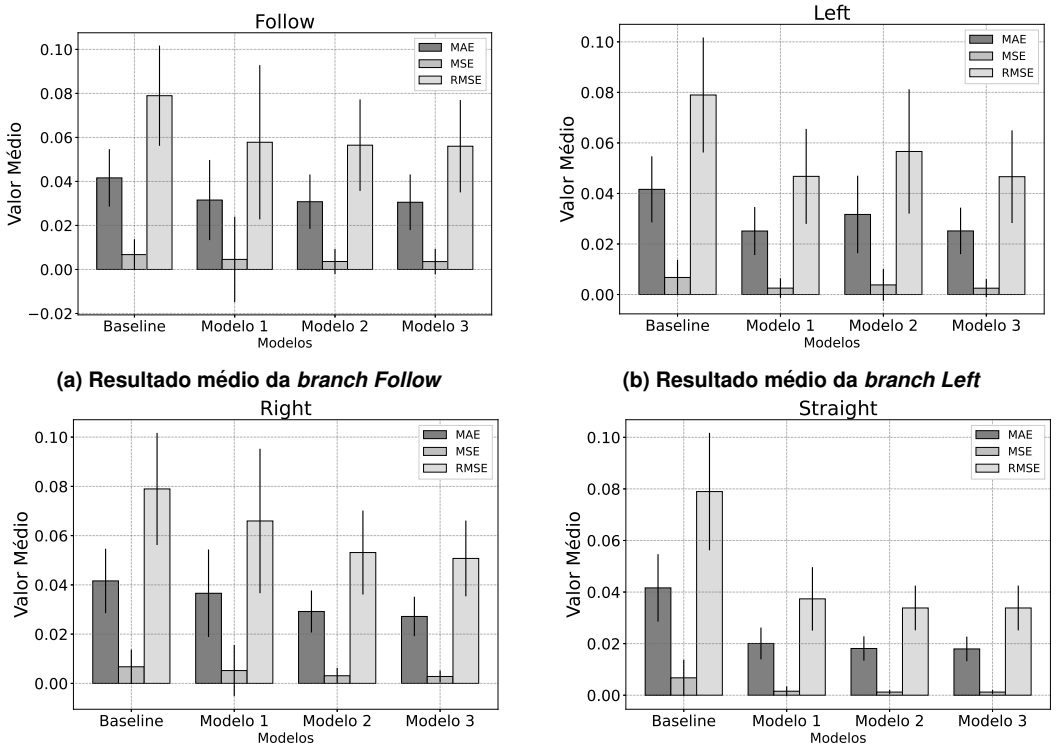
A avaliação do desempenho dos modelos em relação aos erros de condução pode ser efetuada por meio das métricas de precisão, tais como o MSE e o RMSE. Contudo, o RMSE isoladamente não é um bom indicador do erro de localização e o MSE não é eficaz para avaliar o desempenho dos AVs [Codevilla et al. 2018a]. O MSE possui baixa correlação com a qualidade da condução, sendo o MAE identificado como a métrica de melhor correlação [Junior et al. 2021]. Deste modo, na avaliação da precisão dos modelos, foram considerados o MSE e RMSE, adicionando o MAE para uma análise mais completa. As métricas estão contidas no intervalo [0, ∞], onde valores mais baixos são indicativos de maior precisão na condução. O erro é calculado entre valor real do ângulo de direção (a) e o valor do ângulo previsto pelo modelo ($\hat{a}$).

4.2. Resultados

A Figura 4 apresenta o valores médios obtidos de MAE, MSE e RMSE ao longo do treinamento dos modelos. Ao analisar os resultados da Figura 4a, é possível notar que os modelos 1, 2 e 3 exibem resultados similares em relação a *branch Follow*, com RMSE médio de aproximadamente 0.0578, 0.0565 e 0.056, respectivamente. Essa observação indica que as técnicas de pré-processamento têm um impacto mínimo no desempenho durante o treinamento. Além disso, o RMSE de todos os modelos se mantém acima de 0.055, uma característica que não é observada em outras *branches*. A partir da análise dos resultados de treinamento, pode-se concluir que a *branch Follow* apresentou um comportamento complexo de predição, que envolve a realização de diversas manobras de mobilidade, como curvas para esquerda, direita e seguir em linha reta.

Na *branch Left*, ao analisar o gráfico da Figura 4b, é possível notar a redução no MSE médio de 62.81% do modelo 3 em comparação ao *baseline*, além de uma diminuição de cerca de 34.12% em relação ao modelo 2 e 1.2% em comparação com o modelo 1. Também é possível visualizar que o MSE médio do modelo 1 é cerca de 17.40% inferior ao modelo 2. Esses resultados demonstram que somente a normalização dos dados de entrada não é o suficiente para uma melhora no treinamento dos modelos baseados no CIL. Esta melhoria ocorre apenas quando a aumentação de dados também é aplicada.

Pode-se observar na Figura 4d que a *branch Straight* obteve os menores valores em relação as métricas avaliadas. O valor de MAE apresentado pelos modelos baseados em CIL é de 0.02. Este valor apresenta uma redução de 25% em comparação ao menor valor de MAE de outras *branches*. Realizando uma análise dos dados de treinamento, foi possível concluir que os cenários de condução presentes na *branch Straight* não sofrem grandes variações de velocidade e ângulo de direção, fatores que simplificam o aprendizado da rede. Ainda conforme a Figura 4d os modelos 2 e 3 possuem um erro médio similar, onde o modelo 3 diminui o MAE em 56.87% em comparação ao *baseline*, 10.56% em relação ao modelo 1 e 0.94% em relação ao modelo 2.



(a) Resultado médio da *branch Follow*

(b) Resultado médio da *branch Left*

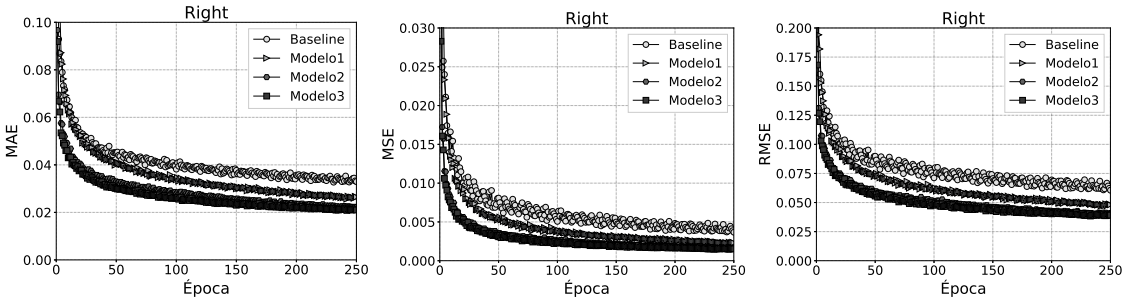
(c) Resultado médio da *branch Right*

(d) Resultado médio da *branch Straight*

Figura 4. Valor médio das métricas ao longo do treinamento

A Figura 5 representa a evolução do MAE, MSE e RMSE ao longo das 250 épocas de treino da *branch Right*. É possível observar nos gráficos 5a, 5b e 5c que o modelo *baseline* demonstra maiores oscilações nos valores ao longo das épocas de treinamento. Essa instabilidade está associada à variabilidade nos dados de entrada do *baseline*, o qual, ao contrário dos demais modelos, é treinado com todo conjunto de dados. Ainda em relação a *branch Right*, na Figura 5a, o modelo 3 atinge uma redução de 34.69% do MAE em relação ao *baseline* e de 25.72% e 6.85% em relação ao modelo 1 e modelo 2,

respectivamente. Portanto, é possível analisar que, nesta *branch*, a normalização auxilia no desempenho do CIL durante o treino dos modelos e este resultado é melhorado quando realizada em conjunto com as técnicas de aumentação de dados.



(a) MAE **(b) MSE** **(c) RMSE**

Figura 5. Evolução das métricas ao longo das épocas - *Right*

A fim de avaliar o desempenho dos modelos no conjunto de teste, a Figura 6 exibe o ângulo de direção predito pelo modelo (em vermelho) e o ângulo real (em azul) no decorrer de 200 *data points* de um arquivo de teste da *branch Left*. Como observado na Figura 6a, o *baseline* não reproduz com precisão a variação no ângulo de direção durante o movimento de curva à esquerda. Em um cenário real, esse comportamento poderia resultar em grande instabilidade na condução. Por sua vez, os modelos 1 e 2, referenciados em 6b e 6c, respectivamente, tendem a estimar ângulos com o mesmo sentido do ângulo real, e podem seguir pequenas variações de direção. No entanto, esses modelos não conseguem prever grandes variações na direção do AV. Por fim, o modelo 3 (Figura 6d), atua de maneira mais precisa, acompanhando até mesmo as grandes variações no ângulo de direção.

Com base na análise das avaliações realizadas, foi possível concluir que o modelo 3 reproduz com maior precisão a variação no ângulo de direção do AV em diferentes cenários de condução. De acordo com os resultados, o maior RMSE médio analisado para o modelo 3 é 0.056, inferior ao menor RMSE médio dos demais modelos avaliados. Esse desempenho, deve-se ao fato deste modelo integrar duas técnicas de pre-processamento ao CIL. Os resultados indicam que a normalização e a aumentação de dados são técnicas complementares para auxiliar na generalização dos modelos.

5. Conclusões e Trabalhos Futuros

Este estudo avaliou modelos de aprendizado destinados a AVs, baseados nas técnicas de IL e CIL. Foram avaliados quatro modelos, com o intuito de analisar o impacto das técnicas de pré-processamento de dados, especificamente a normalização e a aumentação de dados, na capacidade de generalização dos modelos. O modelo *baseline* foi treinado utilizando a estrutura padrão do IL enquanto os demais modelos foram desenvolvidos com base no CIL. No geral, os resultados demonstram que o modelo com normalização e aumentação de dados obteve maior precisão em termos de MAE, MSE e RMSE. Cujo MSE médio apresenta redução de até 62.81% em comparação com os demais modelos. Dessa forma, as técnicas de normalização e aumentação aplicadas em conjunto são uma alternativa para ampliar a performance dos AVs em cenários desconhecidos. Como trabalhos futuros, espera-se desenvolver um modelo CIL capaz de predizer outros sinais de

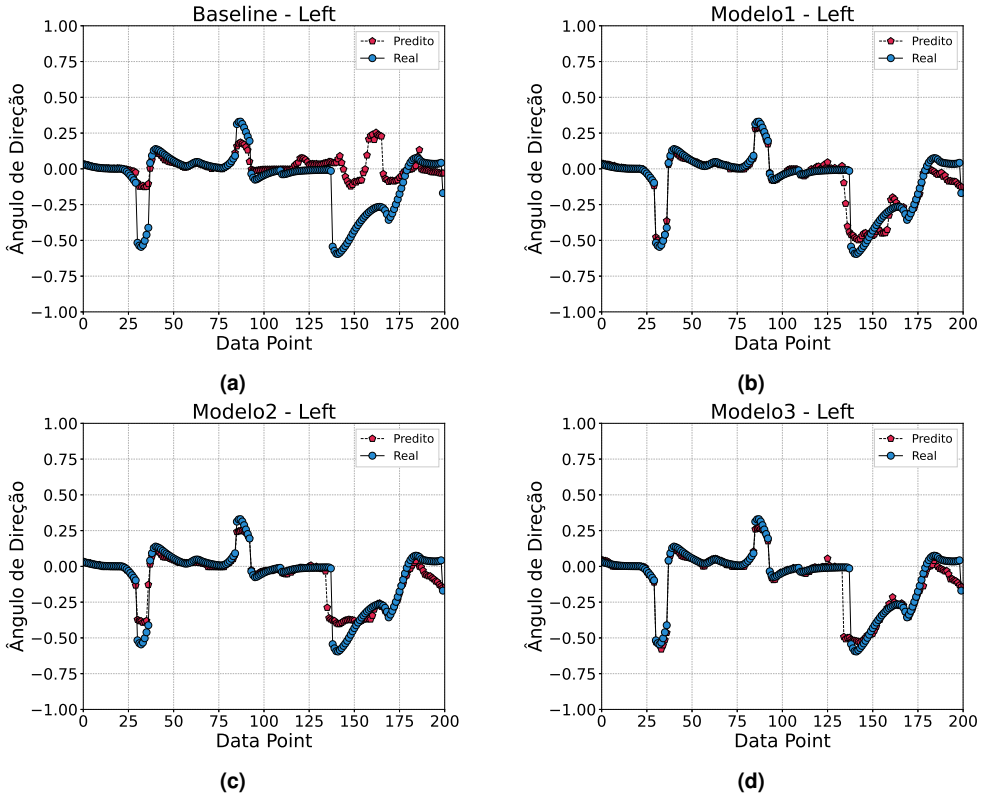


Figura 6. Comparação entre o ângulo de direção predito e real - *Left*

controle de baixo nível, como aceleração e frenagem, trazendo maior controle sob o AV. Além de considerar outras métricas para a avaliação dos modelos em um ambiente de aprendizado federado.

Agradecimentos

Os autores gostariam de agradecer à Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), pelas bolsas #2015/24494-8 e #2019/19105-3. Também à PPI-Softex, com o apoio do MCTI [01245.013778/2020-21].

Referências

Codevilla, F., Lopez, A. M., Koltun, V., and Dosovitskiy, A. (2018a). On offline evaluation of vision-based driving models. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 236–251.

Codevilla, F., Müller, M., López, A., Koltun, V., and Dosovitskiy, A. (2018b). End-to-end driving via conditional imitation learning. In *IEEE International Conference on Robotics and Automation (ICRA)*.

Codevilla, F., Santana, E., Lopez, A. M., and Gaidon, A. (2019). Exploring the limitations of behavior cloning for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., and Koltun, V. (2017). CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*.

Eraqi, H. M., Moustafa, M. N., and Honer, J. (2020). Efficient occupancy grid mapping and camera-lidar fusion for conditional imitation learning driving. In *IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*.

Eraqi, H. M., Moustafa, M. N., and Honer, J. (2022). Dynamic conditional imitation learning for autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 23(12):22988–23001.

Hawke, J., Shen, R., Gurau, C., Sharma, S., Reda, D., Nikolov, N., Mazur, P., Micklethwaite, S., Griffiths, N., Shah, A., and Kndall, A. (2020). Urban driving with conditional imitation learning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*.

Hu, A., Corrado, G., Griffiths, N., Murez, Z., Gurau, C., Yeo, H., Kendall, A., Cipolla, R., and Shotton, J. (2022). Model-based imitation learning for urban driving. In *Advances in Neural Information Processing Systems*, volume 35, pages 20703–20716. Curran Associates, Inc.

Junior, W. L., de Souza, A. M., Cerqueira, E., Rosário, D., and Villas, L. (2021). Mecanismo eficiente de localização cooperativa para veículos autônomos conectados. In *Anais do XXXIX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 336–349. SBC.

Le Mero, L., Yi, D., Dianati, M., and Mouzakitis, A. (2022). A survey on imitation learning techniques for end-to-end autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 23(9):14128–14147.

Ly, A. O. and Akhloufi, M. (2021). Learning to drive by imitation: An overview of deep behavior cloning methods. *IEEE Transactions on Intelligent Vehicles*, 6(2):195–209.

Maharana, K., Mondal, S., and Nemade, B. (2022). A review: Data pre-processing and data augmentation techniques. *Global Transitions Proceedings*, 3(1):91–99.

Teng, S., Chen, L., Ai, Y., Zhou, Y., Xuanyuan, Z., and Hu, X. (2023). Hierarchical interpretable imitation learning for end-to-end autonomous driving. *IEEE Transactions on Intelligent Vehicles*, 8(1):673–683.

Xiao, Y., Codevilla, F., Gurram, A., Urfalioglu, O., and López, A. M. (2022). Multimodal end-to-end autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 23(1):537–547.

Zhang, E., Zhou, H., Ding, Y., Zhao, J., and Ye, C. (2020). Learning how to avoiding obstacles for end-to-end driving with conditional imitation learning. In *Proceedings of the 2019 2nd International Conference on Signal Processing and Machine Learning*. Association for Computing Machinery.

Zheng, B., Verma, S., Zhou, J., Tsang, I. W., and Chen, F. (2022). Imitation learning: Progress, taxonomies and challenges. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–16.