



UNIVERSIDADE FEDERAL DE SANTA CATARINA
TECHNOLOGICAL CENTER
GRADUATION PROGRAM IN SYSTEMS AND AUTOMATION ENGINEERING

Willian do Nascimento Finato Benoski

**A Study on Implicit Behavioral Cloning for Multimodal Behavior Learning in
Autonomous Driving**

Florianópolis
2026

Willian do Nascimento Finato Benoski

**A Study on Implicit Behavioral Cloning for Multimodal Behavior Learning in
Autonomous Driving**

Thesis submitted to the Graduation Program in Systems and Automation Engineering from the Universidade Federal de Santa Catarina as a partial-requisite for obtaining the master degree in Systems and Automation Engineering.
Supervisor:: Prof. Eric Aislan Antonelo, Dr.

Florianópolis
2026

Ficha catalográfica gerada por meio de sistema automatizado gerenciado pela BU/UFSC.
Dados inseridos pelo próprio autor.

Benoski, Willian do Nascimento Finato

A study on Implicit Behavioral Cloning for Multimodal Behavior Learning in Autonomous Driving / Willian do Nascimento Finato Benoski ; orientador, Eric Aislan Antonelo, 2026.

88 p.

Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia de Automação e Sistemas, Florianópolis, 2026.

Inclui referências.

1. Engenharia de Automação e Sistemas. 2. Implicit Behavior Cloning. 3. Autonomous Driving. 4. Multimodality. I. Antonelo, Eric Aislan. II. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Engenharia de Automação e Sistemas. III. Título.

Willian do Nascimento Finato Benoski

**A Study on Implicit Behavioral Cloning for Multimodal Behavior Learning in
Autonomous Driving**

O presente trabalho em nível de mestrado foi avaliado e aprovado, em 26 de novembro de 2025, pela banca examinadora composta pelos seguintes membros:

Prof. Marcelo Ricardo Stemmer, Dr.
Universidade Federal de Santa Catarina

Prof. Paulo Fernando Ferreira Rosa, Ph.D
Instituto Militar de Engenharia

Certificamos que esta é a **versão original e final** do trabalho de conclusão que foi julgado adequado para obtenção do título de mestre em engenharia de automação e sistemas.



Documento assinado digitalmente
Julio Elias Normey Rico
Data: 24/02/2026 05:15:57-0300
CPF: ***.840.859-**
Verifique as assinaturas em <https://v.ufsc.br>

Prof. Julio Elias Normey Rico, Dr.
Coordenador do Programa



Documento assinado digitalmente
Eric Aislan Antonelo
Data: 24/02/2026 11:24:12-0300
CPF: ***.163.409-**
Verifique as assinaturas em <https://v.ufsc.br>

Prof. Eric Aislan Antonelo, Dr.
Supervisor

Florianópolis, 2026.

Aos meus pais e irmã, por acreditarem nesse sonho, aos meus amigos por toda a força nos momentos difíceis e ao meu namorado por todo o suporte.

AGRADECIMENTOS

Agradeço ao meu orientador, Prof. Dr. Eric Aislan Antonelo, pelo suporte técnico, pelas reuniões e orientações necessárias para o desenvolvimento desta dissertação.

Sou também grato ao meu colega e amigo Gustavo Couto, cuja colaboração foi essencial. Ele contribuiu com revisões, programação e com o desenvolvimento da pesquisa.

Agradeço aos meus familiares Sergio, Gislaine e Larissa por todo o apoio e incentivo para que eu enfrentasse esse desafio.

Agradeço à minha amiga Thays pelas diversas conversas, pelos momentos descontraídos e pelos estudos compartilhados.

Por fim, agradeço ao meu namorado, Ray, que, nos momentos de dúvida, me trouxe forças para continuar neste propósito.

RESUMO

A condução autônoma requer tomada de decisão em ambientes complexos e dinâmicos, nos quais podem existir múltiplas ações válidas para um único cenário. Abordagens tradicionais de aprendizado por imitação, como o Behavioral Cloning (BC), apresentam limitações em contextos multimodais, pois tendem a convergir para uma ação média e sofrem com mode collapse. Os Energy-Based Models (EBMs) oferecem uma alternativa ao modelar uma função de energia sobre o espaço de ações, possibilitando o aprendizado de comportamentos mais diversos e adaptáveis. Esta pesquisa investiga o Implicit Behavioral Cloning (IBC), um método baseado em EBM voltado ao aprimoramento do aprendizado de políticas multimodais na condução autônoma. São propostas modificações no algoritmo de treinamento original com o objetivo de aumentar a eficiência do aprendizado da função de energia, resultando no método denominado Data Augmented IBC (DA-IBC). A avaliação do DA-IBC foi conduzida em cenários de condução urbana no simulador CARLA, sob diferentes condições, incluindo a presença ou ausência de rotas pré-determinadas e níveis crescentes de complexidade, com obstáculos dinâmicos como semáforos, pedestres e veículos. O método foi comparado ao IBC e ao BC em tarefas que exigem tomada de decisão multimodal, como interseções. Os resultados indicam que o DA-IBC supera o BC em cenários multimodais. Em interseções com possibilidade de conversão à direita, o agente treinado com BC escolheu essa ação em 94% dos casos, enquanto o DA-IBC distribuiu suas decisões entre seguir à direita (40%), em frente (25%) e à esquerda (35%). Com o aumento da complexidade das tarefas, o DA-IBC apresentou melhor conformidade com os semáforos, reduzindo em 50% as infrações em comparação ao BC, além de aumentar a distância percorrida em 62% em relação ao IBC. Ademais, ao interagir com obstáculos, o DA-IBC evitou a maioria das infrações observadas. Esses resultados sugerem que o DA-IBC constitui uma abordagem promissora para o aprendizado de comportamentos de condução complexos a partir de demonstrações limitadas de especialistas, embora sejam necessárias melhorias adicionais para lidar com cenários que envolvem múltiplos obstáculos simultâneos.

Palavras-chave: Multimodalidade; Modelos baseados em energia; Veículos Autônomos; CARLA; Bird's-Eye View.

ABSTRACT

Autonomous driving requires decision-making in complex and dynamic environments, in which multiple valid actions may exist for a single scenario. Traditional imitation learning approaches, such as Behavioral Cloning (BC), present limitations in multimodal contexts, as they tend to converge to an average action and suffer from mode collapse. Energy-Based Models (EBMs) provide an alternative framework by learning an energy function over the action space, enabling the development of more diverse and adaptable behaviors. This research investigates Implicit Behavioral Cloning (IBC), an EBM-based method aimed at improving the learning of multimodal policies for autonomous driving. Modifications to the original training algorithm are proposed to increase the efficiency of energy function learning, resulting in a method termed Data Augmented IBC (DA-IBC). The evaluation of DA-IBC was conducted in urban driving scenarios within the CARLA simulator under varying conditions, including the presence or absence of predetermined routes and progressively increasing levels of complexity, with dynamic obstacles such as traffic lights, pedestrians, and vehicles. The method was compared against IBC and BC in tasks that require multimodal decision-making, such as intersections. The results indicate that DA-IBC outperforms BC in multimodal scenarios. At intersections with a possible right turn, the BC-trained agent chose this action in 94% of the cases, whereas DA-IBC distributed its decisions among turning right (40%), going straight (25%), and turning left (35%). As task complexity increased, DA-IBC demonstrated improved compliance with traffic lights, reducing infractions by 50% compared to BC, and also increased the distance traveled by 62% relative to IBC. Furthermore, when interacting with obstacles, DA-IBC avoided the majority of observed infractions. These findings suggest that DA-IBC represents a promising approach for learning complex driving behaviors from limited expert demonstrations, although further improvements are required to enhance decision-making in scenarios involving multiple simultaneous obstacles.

Keywords: Multimodality; Energy-Based Models; Implicit Behavior Cloning; CARLA; Bird's-Eye View.

RESUMO EXPANDIDO

1. INTRODUÇÃO

A condução autônoma envolve a tomada de decisão em ambientes complexos e dinâmicos, nos quais o agente deve lidar com semáforos, veículos, pedestres e interseções que admitem múltiplas ações plausíveis. Essa característica multimodal representa um desafio significativo para métodos tradicionais de aprendizado por imitação. Embora essas abordagens tenham demonstrado resultados relevantes no treinamento de veículos autônomos a partir de dados sensoriais brutos, suas limitações tornam-se evidentes em cenários urbanos mais complexos.

O Behavior Cloning (BC) aprende um mapeamento direto entre entradas sensoriais e ações com base em demonstrações especialistas. Apesar de sua simplicidade e eficiência, o método depende de grandes volumes de dados e sofre com erros acumulativos e sensibilidade a mudanças na distribuição dos estados encontrados durante a execução. Mais criticamente, por aprender uma política determinística, o BC não consegue representar múltiplos comportamentos válidos para uma mesma situação. Em cenários nos quais diferentes decisões são igualmente aceitáveis, o modelo tende a prever uma ação média, o que pode resultar em comportamentos irrealistas e inseguros.

A multimodalidade na condução autônoma manifesta-se em situações como interseções, ultrapassagens e ajustes de velocidade, nas quais diferentes trajetórias ou perfis de aceleração podem ser adequados dependendo do contexto. Quando treinado com demonstrações que incluem múltiplas escolhas possíveis, o BC frequentemente converge para soluções intermediárias, fenômeno conhecido como *mode collapse*. Como consequência, o agente pode executar ações que não correspondem a nenhuma das alternativas válidas observadas, comprometendo o desempenho e a segurança.

Como alternativa, os *Energy-Based Models* (EBMs) oferecem uma formulação mais flexível para capturar distribuições multimodais. Em vez de prever diretamente uma ação, o modelo aprende uma função de energia sobre o espaço de ações, atribuindo valores mais baixos às ações desejáveis e mais altos às subótimas. Essa abordagem permite que múltiplas ações válidas recebam baixa energia simultaneamente, preservando a diversidade comportamental sem forçar uma média artificial entre decisões conflitantes. Além disso, o treinamento contrastivo favorece uma melhor generalização e maior capacidade de discriminar ações inadequadas.

Com base nessa formulação, o *Implicit Behavioral Cloning* (IBC) introduz uma política implícita na qual a rede recebe tanto a percepção do ambiente quanto uma ação candidata, retornando a energia associada a essa ação. Neste trabalho, investigamos a aplicação do IBC em um ambiente urbano simulado, utilizando representações do cenário em visão superior e comandos contínuos de controle. Diante das limitações observadas na implementação padrão do método em cenários multimodais, propomos uma modificação no processo de treinamento, denominada **Data-Augmented IBC** (DA-IBC), com o objetivo de aprimorar a representação de múltiplos comportamentos válidos e melhorar o desempenho do agente em ambientes urbanos complexos.

2. OBJETIVOS

O *Behavior Cloning* (BC) aprende políticas de direção por meio de regressão supervisionada a partir de demonstrações especialistas, porém sua formulação explícita e determinística limita a capacidade de generalização e de representação de comportamentos multimodais em cenários urbanos complexos. Em ambientes reais, múltiplas ações podem levar ao sucesso da tarefa, como escolher rotas alternativas em situações de congestionamento ou adaptar a condução conforme o fluxo de tráfego e a presença de agentes dinâmicos. Nesse contexto, o agente precisa decidir entre diferentes trajetórias viáveis para alcançar o mesmo destino, considerando fatores como densidade do tráfego, condições da via e interação com outros veículos e pedestres. Essa natureza multimodal impõe desafios que o BC tradicional não consegue tratar de forma adequada.

Diante dessas limitações, esta dissertação investiga uma modificação na forma de representação da política, explorando o uso do *Implicit Behavioral Cloning* (IBC) como alternativa para modelar decisões multimodais em condução autônoma. O estudo busca analisar a aplicabilidade do IBC em tarefas de direção urbana, comparando seu desempenho com o BC tradicional e avaliando sua robustez à medida que o ambiente simulado se torna progressivamente mais complexo. Além disso, examina-se a capacidade do agente de selecionar diferentes rotas válidas diante de uma mesma interseção, evidenciando sua compreensão da multimodalidade no processo decisório.

O objetivo principal consiste em implementar e integrar o método IBC em um ambiente de simulação de condução autônoma, avaliando seu desempenho em tarefas com rotas fixas e dinâmicas. Entre os objetivos específicos estão o desenvolvimento de arquiteturas neurais adequadas, o treinamento e avaliação do agente em cenários urbanos com diferentes níveis de complexidade, a proposição de melhorias no treinamento e inferência culminando no método *Data-Augmented IBC* (DA-IBC) e a comparação sistemática com outros métodos, incluindo o BC tradicional. Por fim, busca-se avaliar a capacidade do método aprimorado de lidar com cenários multimodais, tomando decisões diversas e válidas em contextos idênticos.

3. METODOLOGIA

A metodologia desta pesquisa foi estruturada em fases bem definidas, combinando investigação teórica e experimentação prática, com o objetivo de desenvolver, implementar e avaliar um agente de condução autônoma baseado em *Implicit Behavioral Cloning* (IBC). Inicialmente, foram consolidados os fundamentos teóricos por meio da análise de disciplinas de pós-graduação e de uma revisão bibliográfica focada em aprendizado por imitação, *Behavior Cloning*, *Energy-Based Models* e técnicas de otimização sem derivadas. Esse processo permitiu a escolha das abordagens metodológicas mais adequadas ao problema de tomada de decisão multimodal em ambientes urbanos.

Para complementar a formação teórica, foram realizados cursos externos voltados a aprendizado de máquina, com ênfase em aprendizado por imitação, além de capacitações em programação Python e sistemas Linux. Essas competências foram essen-

ciais para o desenvolvimento, adaptação e depuração dos algoritmos de condução autônoma empregados ao longo do trabalho, garantindo maior eficiência no processo experimental e na análise dos resultados obtidos.

A fase experimental foi conduzida utilizando o simulador CARLA, uma plataforma de código aberto voltada à pesquisa em condução autônoma, baseada no Unreal Engine. O simulador oferece ambientes urbanos realistas, com suporte a rotas definidas ou dinâmicas, agentes móveis como veículos e pedestres, semáforos, sinalização viária e condições ambientais configuráveis. O agente desenvolvido opera a partir de uma representação em **Bird's-Eye View** (BEV), abstraindo a geometria da cena e os participantes do tráfego, o que facilita o aprendizado mantendo informações espaciais relevantes. A implementação e integração do agente com o simulador foram realizadas em Python, utilizando a API do CARLA e o ambiente de desenvolvimento Visual Studio Code.

Os experimentos foram conduzidos a partir de um conjunto de dados especialistas gerado especificamente para este trabalho, abrangendo cenários com complexidade crescente, desde rotas fixas até ambientes urbanos dinâmicos com múltiplos obstáculos. A avaliação do desempenho considerou métricas como taxa de conclusão de rotas, número de colisões, respeito à sinalização e capacidade de tomada de decisão multimodal. Para viabilizar os experimentos, foram utilizados recursos computacionais locais e infraestrutura de alto desempenho da Universidade Federal de Santa Catarina, além de apoio financeiro fornecido pela FAPESC, permitindo dedicação integral ao desenvolvimento e validação da abordagem proposta.

4. RESULTADOS

A seção de experimentos e resultados apresenta a avaliação comparativa entre três abordagens de aprendizado por imitação para condução autônoma: *Behavioral Cloning* (BC), *Implicit Behavioral Cloning* (IBC) e a proposta *Data-Augmented IBC* (DA-IBC). Os experimentos foram conduzidos no simulador CARLA, utilizando dados coletados no ambiente *Town01* para treinamento e avaliando a generalização no ambiente *Town02*. O conjunto de dados especialista foi gerado por um agente determinístico baseado em planejador topológico e controle PID, resultando em mais de 21 mil amostras rotuladas contendo pares estado-ação. As observações foram representadas por imagens **Bird's-Eye View** (BEV) com múltiplos canais semânticos e temporais, incluindo informações de via, faixas, rota, veículos, pedestres e semáforos.

O primeiro experimento avaliou o comportamento dos agentes em um cenário com rota predeterminada e presença de semáforos, mas sem obstáculos dinâmicos. Nessa configuração, BC e DA-IBC apresentaram desempenho semelhante em termos de respeito aos sinais de trânsito, embora o BC tenha alcançado maiores distâncias percorridas. O DA-IBC demonstrou aprendizado mais rápido nas fases iniciais, porém apresentou comportamento conservador em algumas interseções, especialmente ao retomar o movimento após o sinal verde. A análise das paisagens de energia indicou que o modelo baseado em EBM aprendeu corretamente a restringir ações fora da rota definida.

No segundo experimento, a rota foi removida da entrada do agente, criando um cenário inerentemente multimodal em que múltiplas decisões eram plausíveis nas interseções. Nesse contexto, BC obteve maiores pontuações iniciais devido à simplicidade de seu objetivo aprender a ação média enquanto IBC e DA-IBC exigiram mais tempo de treinamento por modelarem uma função de energia sobre o espaço de ações. Entretanto, apenas IBC e DA-IBC foram capazes de representar distribuições multimodais, permitindo decisões distintas sob as mesmas condições iniciais.

O teste específico de multimodalidade confirmou essa diferença estrutural entre os métodos. O BC apresentou comportamento praticamente determinístico, repetindo a mesma trajetória para um mesmo ponto inicial. O IBC demonstrou alguma variabilidade, porém com viés acentuado para determinadas direções. Já o DA-IBC produziu comportamentos diversos e coerentes, distribuindo decisões entre seguir em frente, virar à esquerda ou à direita, mantendo conformidade com as regras de trânsito. Simulações prolongadas mostraram que apenas o DA-IBC conseguiu explorar grande parte do mapa sem orientação de rota, evidenciando aprendizado efetivo de múltiplos modos de condução.

A análise das funções de energia reforçou esses resultados. Em interseções ambíguas, a paisagem de energia aprendida pelo DA-IBC apresentava múltiplos mínimos locais correspondentes a diferentes ações válidas, permitindo amostragem estocástica coerente via DFO adaptado. Em situações unimodais, como parada em sinal vermelho, a função apresentava mínimo único em aceleração e esterçamento nulos, demonstrando que o modelo é capaz de representar tanto decisões ambíguas quanto determinísticas.

No experimento final, foram introduzidos veículos e pedestres, aumentando significativamente a complexidade do ambiente. Nessa configuração, todos os métodos apresentaram redução na distância percorrida, refletindo maior cautela diante de obstáculos dinâmicos. O BC alcançou maiores distâncias, porém frequentemente às custas de infrações, especialmente em semáforos. O IBC apresentou dificuldades recorrentes, incluindo paradas excessivas. O DA-IBC mostrou comportamento mais conservador e seguro, com menor taxa de colisões relativas e maior respeito às regras, embora frequentemente interrompesse a condução após situações ambíguas.

De forma geral, os resultados indicam que o DA-IBC equilibra multimodalidade e segurança de maneira mais eficaz do que BC e IBC padrão. Enquanto BC apresenta desempenho forte em cenários determinísticos, ele não é capaz de representar múltiplas estratégias plausíveis. O IBC tradicional sofre com limitações na geração de contraexemplos e estabilidade de aprendizado. Já o DA-IBC, ao modificar o processo de amostragem de contraexemplos e o procedimento de inferência, demonstrou maior capacidade de modelar distribuições complexas de ações, favorecendo decisões diversas, consistentes e mais alinhadas ao comportamento especialista em ambientes urbanos dinâmicos.

5. CONSIDERAÇÕES FINAIS

A análise qualitativa dos experimentos revelou limitações recorrentes no comportamento dos agentes, especialmente em cenários com semáforos. Infrações ocorreram

principalmente quando o sinal mudava para vermelho com o veículo já muito próximo da interseção, sugerindo deficiência na antecipação temporal e ausência de planejamento explícito de curto prazo. Em ambientes com maior densidade de elementos dinâmicos, o agente baseado em DA-IBC apresentou episódios de indecisão, como paradas no meio da interseção ou permanência estática após conflitos de prioridade.

Em comparação, o BC demonstrou tendência a desrespeitar sinais de trânsito mesmo após treinamento prolongado, atravessando interseções sem a devida parada. O DA-IBC, por sua vez, mostrou-se mais cauteloso em relação a veículos, porém apresentou falhas na consideração de pedestres, especialmente em faixas de travessia. Ambos os métodos exibiram comportamento subótimo quando confrontados simultaneamente com múltiplas restrições, como um veículo cruzando a via sob sinal verde, resultando em paradas prolongadas e retomadas tardias do movimento.

Esses padrões indicam dificuldades na integração de múltiplas fontes de informação e na tomada de decisão sob incerteza. Em situações nas quais sinais conflitantes estavam presentes como semáforo verde e veículo ainda atravessando a interseção os agentes frequentemente demonstraram indecisão, refletindo limitações na interpretação de prioridade. Tais resultados evidenciam a necessidade de mecanismos mais robustos de raciocínio contextual e coordenação entre percepção e controle.

Como trabalhos futuros, destaca-se a importância de uma avaliação quantitativa mais aprofundada do DA-IBC, incluindo análises segmentadas por tipo de cenário, estudo detalhado de falhas próximas a eventos críticos e testes de consistência comportamental em múltiplas execuções. A realização de testes estatísticos e estudos de ablação sobre estratégias de aumento de dados, regularização da função de energia e procedimentos de inferência também é fundamental para compreender o impacto de cada componente do método e orientar aprimoramentos.

Em síntese, o trabalho demonstrou que a utilização de **Energy-Based Models** no contexto de **Implicit Behavioral Cloning** permite tratar a multimodalidade inerente à condução autônoma de forma mais expressiva do que abordagens tradicionais. A proposta DA-IBC aprimorou o processo de inferência e a geração de contraexemplos, alcançando desempenho comparável ao BC em métricas quantitativas, sem comprometer a segurança, e preservando diversidade comportamental. Apesar das limitações observadas em cenários complexos, o DA-IBC representa um avanço promissor rumo a políticas de condução autônoma mais robustas, seguras e adaptáveis.

Palavras-chave: Multimodalidade; Modelos baseados em energia; Veículos Autônomos; CARLA; Bird's-Eye View.

LIST OF FIGURES

Figure 1 – The image illustrates a multimodal scenario. In (a) a bifurcation is presented to the agent; in (b) the expected behavior of the BC agent; (c) presents the expected behavior for a multimodal method, the agent can choose between both paths.	20
Figure 2 – This figure illustrates the main difference between inputs and outputs in explicit and implicit policy applied to autonomous driving. <i>Top</i> - Explicit BC receives as input the image and returns the decided steering and acceleration; <i>Bottom</i> - On the other hand, Implicit BC receives two inputs: the image and an action, and outputs the energy associated to that action.	21
Figure 3 – Feature map: The blue squares represent the input image matrix (7x7), the kernel is defined by the gray square (3x3), and the green feature map is the result.	31
Figure 4 – CNN Architecture: This figure present a CNN with two convolutional layers with multiple feature maps, and two pooling layers. In the following layers the feature maps are flattened and go to a fully connected layers, pass through a softmax to output the class probabilities. . . .	32
Figure 5 – Camera images : (a) is the camera image from the left side of the car, (b) is the frontal camera image and (c) is the right camera image. The cameras capture images from the simulator with its informations of road, lanes and obstacles.	34
Figure 6 – BEV Representation : (a) contain the desired route in light gray, drivable area in dark gray and lane boundaries in purple. Additionally, (b) represent the nearby pedestrians, (c) the nearby vehicle and (d) the traffic light states, the different time steps of (b) and (c) are represented by the colors red, green and blue, being red the most recent time step and blue the least, (b) (c) and (d) time step are encoded into separate BEV channels.	34
Figure 7 – IBC architecture : Architecture of the energy-based CNN policy. Spatial features extracted from the input image are combined with vehicle control states (previous throttle, steering, brake, gear, and velocity) and a proposed action (acceleration and steering) to produce the corresponding energy value.	44
Figure 8 – Top-view image of CARLA's Town01 the dots are waypoints that define the route, the route start in the light-yellow and end at the red dot.	56

Figure 9 – Top-view image of CARLA’s Town02 the dots are waypoints that define the route, the route start in the light-yellow and end at the red dot.	56
Figure 10 – Training curves of BC and DA-IBC over 300 epochs, trained on Town01 and evaluated on Town02, reporting mean and standard deviation of the distance-based score with traffic light penalty, computed from multiple evaluations every 10 epochs.	61
Figure 11 – Training curves of BC and DA-IBC over 300 epochs, trained on Town01 and evaluated on Town02, reporting mean and standard deviation of the distance traveled of the agents, computed from multiple evaluations every 10 epochs.	62
Figure 12 – Training progress of BC and DA-IBC over 300 epochs, trained on Town01 and evaluated on Town02, reporting mean and standard deviation of the traffic light compliance ratio $\frac{R_{tl}}{E_{tl}}$ (where E_{tl} , computed from multiple evaluations every 10 epochs.	63
Figure 13 – Left: BEV scene perceived by the agent. Right: energy landscape $E(s, a)$ over steering and acceleration, with the zero-steering line indicated in white. Top: agent stopped at a red traffic light; bottom: agent accelerating and turning left after the light turns green. Cross markers denote actions inferred by ten DFO runs for the same BEV input.	64
Figure 14 – Training curves of BC, IBC, and DA-IBC on Town01 with evaluation on Town02, showing the mean and standard deviation of the distance-based score with traffic-light penalty, evaluated every 10 epochs during offline training.	65
Figure 15 – Trajectory of the BC agent in a 3,000 s simulation in Town02, showing limited exploration and lack of multimodal behavior without route guidance, resulting in confinement to a small city region.	68
Figure 16 – Trajectory of the DA-IBC agent during a 3,000 s simulation in Town02, demonstrating learned multimodal behavior that enables diverse decisions at intersections and full city exploration without route guidance.	68
Figure 17 – Left: BEV scene perceived by the agent. Right: energy landscape $E(s, a)$ over steering and acceleration in a multimodal intersection scenario, with zero steering indicated. Top and bottom illustrate left and right turn solutions at the same intersection, with white points showing actions inferred from multiple DFO runs.	69

Figure 18 – Energy $E(s, a)$ as a function of the two-dimensional action a (right plot), in situations where the steering action distribution is nearly unimodal: stopping at a red light. The ten DFO inferences are at (0,0) for acceleration/steering with the fixed BEV as input (left plot).	69
Figure 19 – Training curves of BC, IBC and DA-IBC on Town01 with evaluation on Town02, reporting mean and standard deviation of the normalized distance-based score weighted by traffic-light and collision penalties, evaluated every 10 epochs.	71
Figure 20 – Normalized comparison of traveled distances for BC, IBC and DA-IBC agents in the same training and evaluation conditions previously determined.	72
Figure 21 – Comparison of traffic light compliance ratios for BC, IBC and DA-IBC agents, for 600 epochs of training progress, the traffic light factor is the number of respected traffic lights divided by the total of traffic lights encountered.	73
Figure 22 – Vehicle Collision Score considering the number of collision with vehicles divided by the distance traveled, for BC, IBC and DA-IBC for 600 epochs of training progress.	74
Figure 23 – Pedestrian Collision Score considering the number of collision with vehicles divided by the distance traveled, for BC, IBC and DA-IBC for 600 epochs of training progress.	74
Figure 24 – Top left: BEV image perceived by the agent, while other dynamic obstacles are near (pedestrians and vehicles). Top right: energy landscape $E(s, a)$ of the agent interacting with the environment, where the gradient goes from red (high energy - less likely) to blue (low energy - most likely). Bottom-left: BEV image of the same agent 10 steps further, passing the traffic light. Bottom-right: Energy function of the agent deciding to turn right.	77
Figure 25 – Top left: BEV image of the agent in the road, while a pedestrian is in the road. Top right: Energy function for the agent in a safe distance from the pedestrian still accelerating. Bottom-left: BEV image of the same agent 2 steps further, closer to the pedestrian. Bottom-right: Energy function of the agent braking to avoid colliding, the "x" mark, is in the point of zero steering and negative acceleration.	78
Figure 26 – Similar situation and behavior from 25, but now the dynamic agent is a vehicle.	78

LIST OF TABLES

Table 1 – Comparison of major imitation learning paradigms for autonomous driving.	53
Table 2 – Behavior comparison of trained BC, IBC, and DA-IBC agents in Town02, reporting the frequency of path choices at two-option intersections over multiple episodes from 10 different starting points.	66
Table 3 – BC evaluation results over 400 training epochs, reporting per-episode collisions with vehicles and pedestrians, red-light infractions, and distance traveled across 10 episodes.	75
Table 4 – IBC evaluation results over 520 training epochs, reporting per-episode collisions with vehicles and pedestrians, red-light infractions, and distance traveled across 10 episodes.	75
Table 5 – DA-IBC evaluation results over 4560 training epochs, reporting per-episode collisions with vehicles and pedestrians, red-light infractions, and distance traveled across 10 episodes.	76
Table 6 – Mean value of tables 5, 4 and 3 and result of the score for the calculated means	76

CONTENTS

1	INTRODUCTION	19
1.1	MOTIVATION	21
1.2	RESEARCH QUESTIONS	22
1.3	OBJECTIVES	22
1.3.1	Main Objective	22
1.3.2	Specific Objectives	23
1.4	DOCUMENT STRUCTURE	23
2	BACKGROUND	25
2.1	MACHINE LEARNING	25
2.1.1	Supervised and Unsupervised Learning	26
2.1.2	Reinforcement Learning	27
2.1.3	Imitation Learning	28
2.2	CONVOLUTIONAL NEURAL NETWORKS	29
2.2.1	Introduction	29
2.2.2	Convolution Operation	29
2.2.3	CNN Architecture	31
2.2.4	Adam optimizer	32
2.3	BIRD'S-EYE VIEW REPRESENTATION	33
2.4	BEHAVIOR CLONING	35
2.4.1	Introduction	35
2.4.2	Mathematical Formulation	36
2.5	ENERGY-BASED MODELS	37
2.5.1	Introduction	37
2.5.2	Training	38
2.6	KERNEL DENSITY ESTIMATION	38
2.7	IMPLICIT BEHAVIORAL CLONING	39
2.7.1	Training	40
2.7.2	Advantages of IBC	40
2.7.3	Inference using Derivative-Free Optimization (DFO)	41
2.7.4	IBC Architecture	42
3	METHODOLOGY	45
3.1	METHODOLOGICAL PROCEDURE	45
3.1.1	Courses	45
3.1.2	CARLA Simulator	45
3.2	RESOURCES	46
4	RELATED WORK	48
4.1	BEHAVIOR CLONING	48

4.2	IMPLICIT BEHAVIOR CLONING AND ENERGY-BASED MODELS	49
4.3	DIFFUSION MODELS	51
4.4	GENERATIVE ADVERSARIAL IMITATION LEARNING (GAIL)	52
4.5	COMPARISON OF METHODS	53
5	EXPERIMENTS AND RESULTS	55
5.1	EXPERT AND AGENT	55
5.1.1	Expert Dataset	55
5.1.2	Image input	57
5.1.3	Action Representation	58
5.1.4	Data Augmented IBC (DA-IBC)	58
5.1.5	Adapted Derivative-Free Optimization (DFO)	59
5.2	EXPERIMENTS	59
5.2.1	Traffic lights with predetermined route	60
5.2.2	Traffic lights without predetermined route	63
5.2.3	Multimodality test	65
5.2.4	Vehicles and pedestrians	70
6	CHALLENGES AND FUTURE WORKS	80
7	CONCLUSION	82
	REFERENCES	84

1 INTRODUCTION

Autonomous driving encompasses decision making in complex and dynamic environments, presenting obstacles such as traffic lights, vehicles, pedestrians, and intersections with multiple plausible actions, that can be taken in each different scenario. This inherent multimodality challenges traditional learning approaches such as Behavior Cloning (BC) (POMERLEAU, 1989).

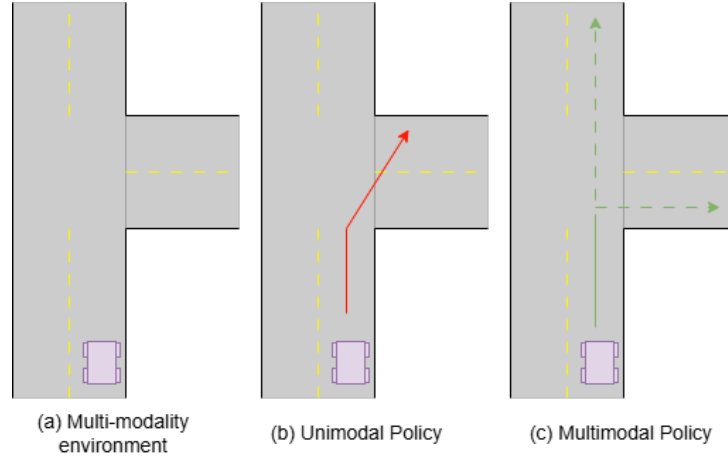
BC aims to learn a direct mapping from sensory inputs to actions based on human demonstrations. The method successfully achieved good results in training an agent capable of controlling a self-driving car named ALVINN only using raw sensor data as input (POMERLEAU, 1989). However, BC requires a large and complex expert dataset to be able to achieve good results, it also suffers from cascading errors and sensitivity to covariate shifts, since the models are trained on a subset of necessary samples to achieve a robust driving, but most importantly, BC is incapable to represent multiple valid behaviors (COUTO; ANTONELO, 2025). In contrast, Energy-Based Models (EBMs), as stated by LeCun, Chopra, *et al.* (2006), offer a more flexible set of tools that can capture multimodal behaviors, leading to more robust and generalizable autonomous agents.

Multimodality in autonomous driving refers to the existence of multiple feasible trajectories or control actions that a driver may take in a given situation. For instance, in an intersection, a vehicle may turn left, right, or proceed straight depending on various factors such as traffic flow and pedestrian movement (CODEVILLA *et al.*, 2018); a driver can choose different timings or acceleration patterns when overtaking a slower vehicle (BANSAL; KRIZHEVSKY; OGALE, 2018); depending on comfort and aggressiveness, a driver may brake early and gradually or wait and decelerate sharply (CHAI *et al.*, 2019).

Traditional BC struggles to model this variability, as it typically learns a deterministic policy that outputs a single action per input. When multiple actions are present in the training data, BC tends to regress to an average of these actions, leading to mode collapse, a behavior that fails to reflect realistic decision-making (RHINEHART; MCALLISTER; LEVINE, 2018). In Fig. 1 (a) a road bifurcation is presented to the agent, with the possibility to go straight or turn right, where the route is not predefined, as for Fig. 1 (b), the BC agent faces the bifurcation, when trained on both left and right turns, BC may predict an intermediate steering action, leading the agent to drive off the road and consequently to failure.

Energy-Based Models (EBMs) provide an alternative approach that better captures the multimodal nature of driving decisions. Instead of directly regressing an action, EBMs learn an energy function over the action space, assigning lower energy to desirable actions and higher energy to suboptimal ones. EBMs naturally allow for multiple

Figure 1 – The image illustrates a multimodal scenario. In (a) a bifurcation is presented to the agent; in (b) the expected behavior of the BC agent; (c) presents the expected behavior for a multimodal method, the agent can choose between both paths.



valid actions by assigning low energy to all feasible actions rather than forcing a single output, which is represented in Fig. 1 (c). This enables the model to learn a more expressive representation of human-like driving behavior without averaging conflicting actions.

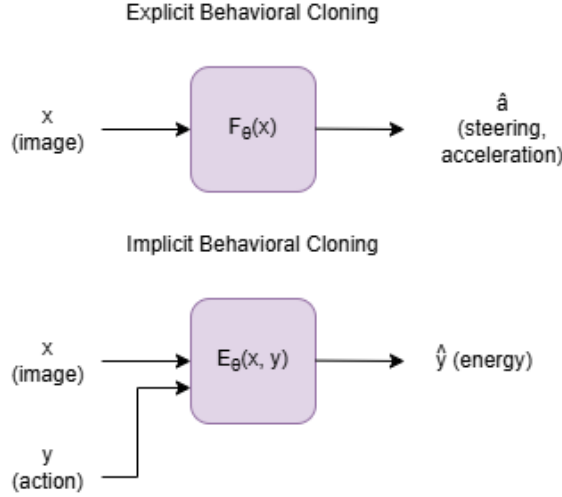
Unlike BC, which minimizes a supervised loss on expert demonstrations, EBMs are trained contrastively: they learn to assign low energy to expert actions and high energy to non-expert ones. This improves generalization and allows EBMs to recognize and avoid unrealistic actions. At inference time, EBMs can be queried with multiple candidate actions, selecting the one with the lowest energy. This dynamic decision-making process allows for sampling from a multimodal action distribution (FLORENCE *et al.*, 2022), making EBMs inherently more adaptable to varying driving conditions.

Implicit Behavioral Cloning (IBC) was proposed by (FLORENCE *et al.*, 2022), where an EBM receives an action input in addition to sensory perception (usually images), the agent then learns an energy function and outputs a corresponding energy value (Fig. 2), explaining the implicit terminology. As for the explicit models, the policy returns directly as output the action to be taken. IBC was shown success in robotics in that work in multimodal scenarios, but its standard implementation has some limitations for other applications such as autonomous driving.

In this work, we investigate the ability of Implicit Behavioral Cloning (IBC) to model multimodal action distributions in an urban driving environment simulated in CARLA, an autonomous driving simulator that closely replicates real-world conditions. The agent receives a Bird's-Eye View (BEV) representation of the scene as input and outputs continuous control commands, namely steering and acceleration (ANTONELO *et al.*, 2024).

Experimental results show that the standard IBC agent repeatedly selected the

Figure 2 – This figure illustrates the main difference between inputs and outputs in explicit and implicit policy applied to autonomous driving. *Top* - Explicit BC receives as input the image and returns the decided steering and acceleration; *Bottom* - On the other hand, Implicit BC receives two inputs: the image and an action, and outputs the energy associated to that action.



same action in situations with multiple valid paths. To address this limitation, we adapt the IBC training algorithm to improve the learning of the energy function and to modify the strategy used for sampling counter-examples. We refer to this enhanced approach as Data-Augmented IBC (DA-IBC). The proposed modification significantly improves the agent's ability to represent and execute multimodal behaviors in complex urban environments.

The investigation starts by evaluating BC, IBC and DA-IBC for their ability to respect traffic lights in different towns with route, followed by removing the route to check the behavior of the agent in multimodal situations while dealing with traffic lights. To enhance the complexity of the environment and better represent common real scenarios, dynamic obstacles are added, as pedestrians and vehicles. Also, the multimodality of each method is evaluated by presenting the same intersection to the agent repeatedly.

1.1 MOTIVATION

Behavior Cloning (BC) learns driving policies via supervised regression on expert demonstrations, but its explicit and deterministic action formulation limits its ability to generalize and represent multimodal behaviors in complex urban driving scenarios. Urban driving presents a wide array of complex scenarios in which multiple plausible actions can lead to successful task completion. For example, during traffic congestion, human drivers often opt to reroute to avoid delays or choose paths with less intense traffic. Although autonomous vehicles are increasingly capable of operating within such environments, their ability to deviate from pre-defined routes and dynamically recal-

culate alternatives remains underexplored. This challenge introduces a multimodal decision-making context, in which BC is not suitable to achieve a good performance. The agent must choose between several viable paths to reach the same destination, depending on surrounding factors such as traffic density, road conditions, and dynamic agents.

This dissertation aims to investigate an improvement to BC, changing the form of its policy, exploring the capacity of Implicit Behavioral Cloning (IBC) to model and execute effective driving strategies within urban environments, particularly under multimodal conditions. By examining the agent's performance in scenarios where multiple valid outcomes are possible, it is evaluated its ability to adapt and choose different paths while maintaining safe and efficient driving behavior. This research not only contributes to the understanding of IBC in complex decision-making scenarios but also to future work to improve the robustness and generalization of policies in more intricate real-world environments.

1.2 RESEARCH QUESTIONS

This work is guided by the following research research questions:

1. **Applicability of IBC in Autonomous Driving:** Is it feasible to apply the Implicit Behavioral Cloning (IBC) method to autonomous driving tasks and achieve satisfactory results?
2. **Comparison with Traditional Behavioral Cloning:** Can IBC outperform standard BC in terms of driving performance and policy robustness?
3. **Dealing with complex environments:** Can the IBC agent continuously get good performances as the simulated environment progressively approaches the real world?
4. **Multimodal Decision-Making Capability:** Is the IBC agent capable of selecting different valid routes when faced with the same intersection, thus demonstrating an understanding of multimodality in decision-making?

1.3 OBJECTIVES

This chapter outlines the main and specific objectives of the research.

1.3.1 Main Objective

The main objective is to implement and integrate the IBC method for autonomous driving within the CARLA simulator and evaluate its performance on both fixed and dynamic routing tasks, identifying its strengths and limitations.

1.3.2 Specific Objectives

1. Design end-to-end neural network architectures suitable for autonomous driving tasks in the CARLA simulator;
2. Train and evaluate the IBC method across a range of simulated urban scenarios, including:
 - Presence of traffic lights and predetermined route;
 - Presence of traffic lights and removal of predetermined route;
 - Incremental introduction of dynamic elements such as traffic lights, other vehicles, and pedestrians, while the route is removed;
3. Evaluate and improve IBC method proposing alterations in the training and inference, developing the Data Augmented IBC method;
4. Compare the performance of DA-IBC and IBC against baseline and alternative approaches, including traditional BC;
5. Assess DA-IBC's ability to handle multimodal scenarios, specifically its capacity to make diverse yet valid decisions when confronted with the same environmental context.

1.4 DOCUMENT STRUCTURE

- **Chapter 2 – Background:** This chapter introduces the fundamental concepts, theories, and definitions necessary to understand the problem addressed in this work. It provides the technical and theoretical foundations on which the proposed approach is built.
- **Chapter 3 – Methodology:** This chapter describes the proposed methodology in detail. It explains the overall approach, system design, resources, and procedures adopted to address the research problem.
- **Chapter 4 – Experiment and Results:** This chapter presents the experimental setup, evaluation metrics, and datasets used in the study. It discusses the obtained results and provides an analysis of the performance and effectiveness of the proposed methodology.
- **Chapter 5 – Related Works:** This chapter reviews and analyzes existing literature and previous studies related to the topic. It highlights relevant approaches, compares them with the proposed method, and identifies gaps that motivate this research.

- **Chapter 6 – Challenges and Future Works:** This chapter summarizes the main challenges and limitations encountered in the proposed method and discusses their impact on performance. It also outlines potential directions for future research, highlighting possible improvements and next steps to advance the development of the approach.
- **Chapter 7 – Conclusion:** This chapter summarizes the main contributions and findings of the work. It also discusses limitations and suggests possible directions for future research.

2 BACKGROUND

2.1 MACHINE LEARNING

Machine learning (ML) provides automated methods that can detect patterns in the data to perform tasks (FRANÇOIS-LAVET *et al.*, 2018). ML is commonly used to process and detect patterns in large batches of data, which would be impractical to do manually. Bengio, Goodfellow, and Courville (2017) explain that the learning process expects the agent to learn from provided specialized examples for a specific task, is further evaluated and improved as the training progresses.

Li (2017) states that ML is a methodology that uses probability and statistics theories to employ optimization processes. Pattern analysis and decision-making have been key to development in numerous fields: information retrieval, data mining, computer vision, robotics, and autonomous driving have been explored with different ML methods.

There are different tasks for which ML is employed: classification tasks to specify inputs into different categories, such as object recognition; regression tasks, in which with a given input, the agent must predict a numerical value, commonly used in algorithmic trading; transcription enables the agent to recognize textual patterns in images and output the data in textual form; as for translation, the agent gathers and recognizes symbols of a language and outputs the text in another language (BENGIO; GOODFELLOW; COURVILLE, 2017).

ML algorithms consist of four components: an expert dataset composed of different scenarios (states) and the decision (action) that should be performed; the loss function that measures the accuracy of the trained agent; an optimization procedure to reduce training error, consequently improving the results; and a trained model, which is evaluated in different environments (LI, 2017). This process requires continuous testing, with different values of hyperparameters, to achieve the desired behavior.

"Usually we are interested in how well the machine learning algorithm performs on data that it has not seen before" (BENGIO; GOODFELLOW; COURVILLE, 2017), which determines if the agent is able to act in the real world. In order to achieve this, we separate the training set from the test set, during training the agent learns while interacting with the training set, receiving the examples on how to act, and then it is further tested using the test set, in both procedures the accuracy is measured.

The agent's ability to operate effectively on previously unobserved inputs is referred to as generalization, and it constitutes a fundamental challenge in machine learning (BENGIO; GOODFELLOW; COURVILLE, 2017). During training, the model's performance is evaluated by minimizing the training error through an optimization process. However, the primary objective of most machine learning algorithms extends beyond reducing training error; they aim to minimize the test error, also known as the

generalization error, which reflects the model's performance on unseen data.

In certain instances, the trained model fails to exhibit the expected behavior, which is intrinsically linked to the magnitudes of the training and test errors. When the model is unable to minimize the training error, the agent lacks the capacity to perform the task even in previously encountered scenarios, a condition referred to as underfitting. Conversely, a significant discrepancy between training and test errors suggests that the model has overfitted to the training data, effectively capturing scenario-specific patterns without acquiring the generalization capabilities necessary for robust performance in previously unseen environments (BENGIO; GOODFELLOW; COURVILLE, 2017).

ML tasks consists of three main types: supervised learning, unsupervised learning, and reinforcement learning (FRANÇOIS-LAVET *et al.*, 2018). The latter is discussed in greater depth in the subsequent section. The author describes supervised learning as a method that requires labeled inputs to be able to correlate the input information and the output results, through regression or classification. Unsupervised learning, on the other hand, “[...] attempts to extract information from data without labels, e.g., clustering and density estimation” (LI, 2017). The agent classifies the data through the observed characteristics and infers the results.

2.1.1 Supervised and Unsupervised Learning

Machine learning algorithms are typically categorized into supervised and unsupervised learning, depending on the nature of the training data and the task objective. These paradigms define how models extract knowledge from data and are foundational to a wide range of applications across science and industry.

Supervised learning requires a labeled dataset, where each input sample is paired with a corresponding target output. The goal is to learn a function that maps inputs to outputs by minimizing a predefined loss function, typically through iterative optimization techniques. The performance of supervised models is often measured using metrics such as accuracy, precision, recall, or mean squared error, depending on the nature of the task (classification or regression). Because the training process is guided by explicit target labels, supervised learning can achieve high predictive performance when sufficient labeled data is available. Common applications include image classification, speech recognition, and medical diagnosis (BENGIO; GOODFELLOW; COURVILLE, 2017; BISHOP, 2006).

A notable example within the autonomous driving domain and the focus of this work is Implicit Behavioral Cloning (IBC), where an agent learns to imitate expert driving behavior directly from human demonstrations using supervised signals (FLORENCE *et al.*, 2022). This approach has proven effective in scenarios where collecting labeled data is feasible and system behavior can be derived from expert trajectories.

2.1.2 Reinforcement Learning

Reinforcement learning (RL) is a class of machine learning algorithms that enables an agent to learn behaviors through rewards. In RL, the agent interacts with an environment and performs actions that lead to certain outcomes. Positive rewards are given for actions deemed beneficial, while negative rewards are assigned to actions that deviate the agent from the desired goal. The agent's objective is to maximize its cumulative reward, seeking the highest possible value by selecting actions that lead to better outcomes. As the agent interacts with the environment, it progressively learns which actions results in larger rewards, refining its decision-making over time. Importantly, the environment is not pre-defined and must be explored and learned through the agent's actions (SUTTON; BARTO, 2018).

The RL learning process can be classified into two primary settings: offline and online. In offline RL, an expert dataset is used to gather examples of state-action pairs, which the agent then uses to learn a policy. This approach requires that all training data is collected prior to the learning process and is often used in situations where collecting real-time data is expensive or impossible. Conversely, in online RL, the agent continuously interacts with the environment, gathering data as it learns. In this setting, the agent's behavior is updated progressively over time, which enables it to adapt to dynamic environments.

A crucial element of RL is the exploration versus exploitation dilemma, where the agent must balance exploring new actions to discover potentially better outcomes, and exploiting known actions that have previously led to high rewards. As Sutton and Barto (2018) note: "The agent has to exploit what it has already experienced in order to obtain reward, but it also has to explore in order to make better action selections in the future". This dilemma is essential because, while exploration helps the agent gather more information, it might initially result in lower rewards. On the other hand, excessive exploitation might limit the agent's ability to discover higher-reward actions in the long term.

It is important to understand the features that describe the RL problems. The progress of the algorithm is determined by the time step t , the agent receives the state from the environment st in the state space $\mathcal{S}$. The action at is selected from the action space $\mathcal{A}$. The agent behavior is defined by a policy π that correlates st and at . For every action performed, a reward rt is received from the reward function (SUTTON; BARTO, 2018). Once the agent receives a terminal state, that is, the agent is incapable of performing any action, or the goal is achieved.

The process continues until the agent reaches a terminal state, either by being unable to perform any further actions or by achieving the goal. The learning process repeats across multiple episodes until the agent consistently achieves the desired behavior or completes a predefined number of training epochs. Training is considered

successful when the agent reaches a predefined percentage of successful attempts, indicating that it has learned the optimal behavior. If the agent fails to meet this success criterion, training is typically restarted, and hyperparameters may be adjusted to improve performance.

2.1.3 Imitation Learning

Imitation Learning (IL) is a type of machine learning where an agent learns to perform tasks by observing and mimicking the behavior of an expert. The primary objective of IL is to develop agents that can replicate human-like actions, which is particularly useful in situations where it's difficult or even impossible to explicitly define an optimal policy through traditional methods. Instead of programming every interaction or task explicitly, IL allows the agent to learn complex behaviors simply by observing an expert's actions. IL was initially designed with the aim of enabling agents to reproduce human-like behavior in a variety of contexts (ATTIA; DAYAN, 2018).

One of the main strengths of IL is its ability to generalize across different environments, even those that contain states the agent has never seen before. This is especially useful in dynamic settings, such as urban driving, where creating precise mathematical models can be very challenging due to the complexity and variability of the environment. In the case of autonomous vehicles, IL has proven successful by allowing the agent to learn how to drive by observing human drivers, avoiding the need to manually program every possible scenario (ATTIA; DAYAN, 2018). By choosing the most appropriate action based on observed demonstrations, IL helps the agent make decisions in situations it might not have encountered during training.

There are two main approaches in IL: passive learning and active learning. In a passive setting, the agent is given a set of demonstrations from the expert, which typically include sequences of state-action pairs that show the expert's optimal behavior. The agent then learns to replicate these actions, usually through supervised learning, without any interaction with the environment. On the other hand, in an active setting, the agent interacts with the environment and, when necessary, seeks guidance from the expert to determine the best action to take. This approach allows the agent to refine its behavior over time, improving its performance based on both direct experience and expert feedback.

A key distinction within IL is between behavioral cloning and inverse reinforcement learning (IRL). While behavioral cloning directly maps observations to actions, IRL attempts to infer the underlying reward function that the expert is implicitly optimizing. Essentially, IRL focuses on understanding the expert's decision-making process and using this understanding to guide the agent's behavior. Both techniques have been applied successfully across various domains, such as robotics, autonomous vehicles, and healthcare (ROSS *et al.*, 2011; ZIEBART *et al.*, 2008).

IL methods are particularly valuable when the task at hand is difficult to define using traditional programming or reinforcement learning. For example, in autonomous driving, where there are countless possible scenarios, imitation learning offers a more flexible approach. By learning from human drivers, the agent can make decisions in complex and unpredictable environments, something that would be almost impossible to explicitly program. As a result, IL has become a vital tool in fields that require high-level decision-making in uncertain and dynamic environments (POMERLEAU, 1989).

2.2 CONVOLUTIONAL NEURAL NETWORKS

This section presents the fundamental concepts of Convolutional Neural Networks (CNNs), including their applications and underlying mathematical principles. This method plays a crucial role in processing the input images utilized in this study.

2.2.1 Introduction

Convolutional Neural Networks (CNNs) are a class of deep learning architectures particularly well-suited for processing data with a grid-like structure, such as images or temporal signals. They are inspired by the organization of the visual cortex in biological systems, where neurons respond to stimuli only within a restricted region of the visual field known as the receptive field (LECUN; BOTTOU, *et al.*, 1998). This biological inspiration allows CNNs to extract spatial hierarchies of features from input data through local connectivity and parameter sharing.

The fundamental operation in CNNs is the convolution, where a set of learnable filters (or kernels) is convolved over the input data to compute feature maps. Each filter is responsible for capturing specific patterns such as edges, textures, or shapes. As the network depth increases, successive layers learn increasingly abstract representations of the input, enabling robust feature extraction and semantic understanding (BENGIO; GOODFELLOW; COURVILLE, 2017; O'SHEA, 2015).

2.2.2 Convolution Operation

Convolution is the fundamental mathematical operation of Convolutional Neural Networks (CNNs) (BENGIO; GOODFELLOW; COURVILLE, 2017), enabling the automatic extraction of local patterns from structured data such as images. In the context of CNNs, convolution refers to a discrete operation where a small matrix of learnable weights, known as a kernel or filter, is systematically applied across the input data to produce feature maps (BENGIO; GOODFELLOW; COURVILLE, 2017).

These filters are trained to detect specific features such as edges, textures, or more complex shapes. Depending on the learned weights, convolution can perform operations similar to smoothing, sharpening, or edge detection. Unlike classical signal

processing, where filters are fixed and manually designed, CNN filters are learned through backpropagation, which updates the weights to minimize a task-specific loss function (LECUN; BOTTOU, *et al.*, 1998).

Mathematically, convolution involves computing a weighted sum over a local region. It is formally defined using two functions: the input signal x and the kernel or filter w . If both functions are continuous, the convolution is expressed via an integral; if the domain is discrete, as in most CNN applications, a summation is used instead. The general formulation of the convolution operation is given as:

$$s(t) = \int x(a)w(t-a)da = \sum_{a=-\inf}^{\inf} x(a)w(t-a), \quad (1)$$

where $s(t)$ is the resulting feature map, $x(a)$ is the input, and $w(t-a)$ is the weight function, with a being the measurement's age, and t the time step.

In CNNs, the discrete formulation is used, and the operation is typically applied in two or more dimensions. The output, known as the feature map, encodes the presence of specific features at different spatial locations. CNNs work with two-dimensional images, composed of pixels disposed in a grid, where m is the rows and n the columns. The equation becomes:

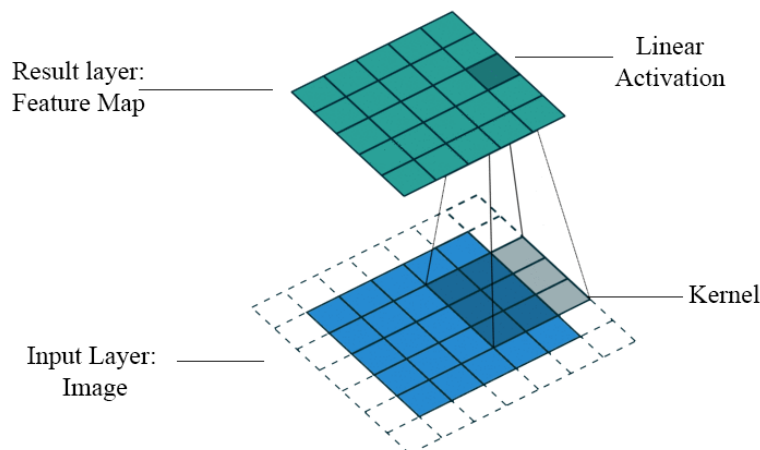
$$s(t) = \sum_m \sum_n I(i-m, j-n)K(m, n), \quad (2)$$

where I is the input image, and K is the kernel. This operation is illustrated in Fig. 3, where the kernel slides across the input matrix, starting at the top left. The sum of the products at each position is passed through an activation function, such as the Rectified Linear Unit (ReLU), resulting in a new pixel value that is then placed in the resultant feature map.

The interaction between the kernel and the input in Convolutional Neural Networks (CNNs) offers several advantages for feature detection in images. Since the kernel is significantly smaller than the input image, it can efficiently capture local features, such as edges, textures, and simple shapes, without the need to process the entire image at once. This localized approach reduces the number of parameters required, lowering memory usage and computational complexity, and leading to faster processing and more efficient training (BENGIO; GOODFELLOW; COURVILLE, 2017).

Moreover, the use of weight parameter sharing in CNNs further improves efficiency. Unlike traditional Artificial Neural Networks (ANNs), where each weight is used only once per connection, CNNs reuse the same set of kernel weights across the entire input. As the kernel slides over different regions of the input (a process known as convolution), the same parameters are applied repeatedly. This reuse of weights not only reduces the total number of learnable parameters but also helps the model generalize

Figure 3 – Feature map: The blue squares represent the input image matrix (7x7), the kernel is defined by the gray square (3x3), and the green feature map is the result.



Source: Mishra D., Towards data science (2019)

better, especially in visual recognition tasks such as image classification and object detection (BENGIO; GOODFELLOW; COURVILLE, 2017).

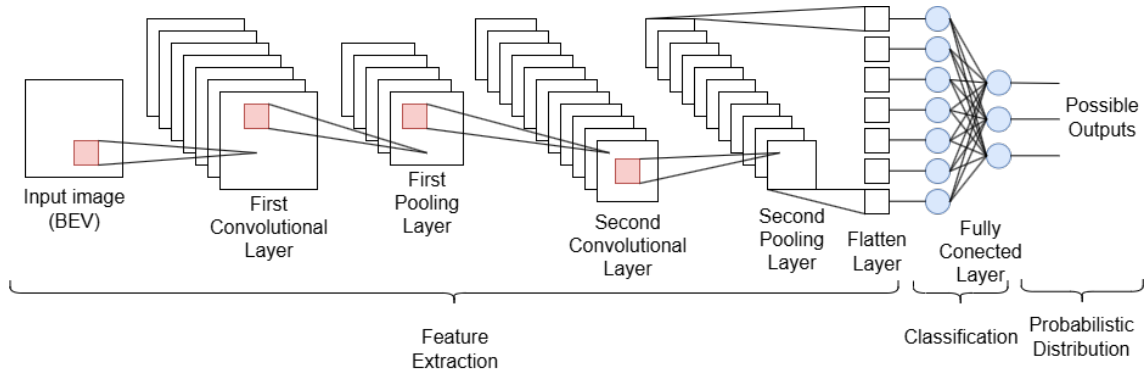
2.2.3 CNN Architecture

A standard Convolutional Neural Network (CNN) architecture is composed of a hierarchical sequence of layers, each contributing to the progressive abstraction of input features. The architecture typically begins with one or more convolutional layers, which apply a set of learnable kernels (filters) to the input via discrete convolution operations. This process generates feature maps that highlight spatially local patterns, such as edges and textures. To introduce nonlinearity and enhance the representational capacity of the model, the output of each convolution is passed through a nonlinear activation function, commonly the Rectified Linear Unit (ReLU), defined as $f(x) = \max(0, x)$ (BENGIO; GOODFELLOW; COURVILLE, 2017).

Following the activation stage, a pooling layer is typically applied to reduce the spatial dimensions of the feature maps. This is achieved via operations such as max pooling or average pooling, which compute a summary statistic over a local neighborhood, introducing translational invariance and reducing computational complexity. Pooling helps the network retain essential information while discarding less relevant variations (BENGIO; GOODFELLOW; COURVILLE, 2017).

Once spatial feature extraction is complete, the resulting multidimensional feature maps are transformed by a flattening layer into a one-dimensional feature vector. This vector is then passed to one or more fully connected layers, which perform classification. These layers act similarly to those in traditional Artificial Neural Networks

Figure 4 – CNN Architecture: This figure present a CNN with two convolutional layers with multiple feature maps, and two pooling layers. In the following layers the feature maps are flattened and go to a fully connected layers, pass through a softmax to output the class probabilities.



(ANNs), where each neuron is connected to all activations from the previous layer. The final output layer typically employs a softmax activation function in classification tasks, converting raw logits into probability distributions over the target classes.

The loss function, such as cross-entropy loss for multi-class classification, is applied at the final layer to quantify the error between predicted and true labels. This loss is minimized using gradient-based optimization algorithms such as stochastic gradient descent (SGD) or Adam. CNNs are suitable for computer vision tasks including image classification, object detection, and scene understanding (LECUN; BOTTOU, *et al.*, 1998; BENGIO; GOODFELLOW; COURVILLE, 2017). This architecture is illustrated in Fig. 4 which summarizes the process of convolution and feature extraction. A kernel slides over the input (a BEV image) and applies convolution operations, producing a feature map. Each kernel produces one feature map, and the combination of multiple feature maps forms the output of the first convolutional layer. A pooling layer then reduces the spatial dimensions of these maps. This process is repeated over n layers. After the final pooling layer, the resulting 2D feature maps are flattened into a 1D array. This array is passed through a fully connected layer for classification. Finally, the outputs are passed through a softmax function to obtain class probabilities.

2.2.4 Adam optimizer

The Adam optimizer (Adaptive Moment Estimation) is one of the most widely used optimization algorithms in training deep learning models, including Convolutional Neural Networks (CNNs). Introduced by Kingma and Ba (2014), Adam combines the advantages of two popular stochastic gradient descent extensions: AdaGrad, which works well with sparse gradients, and RMSProp, which handles non-stationary objectives effectively.

Adam is particularly well-suited for CNNs due to its ability to adapt the learning

rate for each parameter individually, which is useful when training large models with many convolutional and fully connected layers. This makes training more robust and efficient, especially when the architecture includes millions of parameters, as is common in modern CNNs.

Adam maintains exponentially decaying averages of past gradients and their squared values. Given a loss function $\mathcal{L}(\theta)$ and its gradient $\nabla_{\theta}\mathcal{L}(\theta)$ at time step t . The parameters are initialized as follows: θ_0 is the model parameters, $m_0 = 0$ is the first moment vector or the mean of gradients, $v_0 = 0$ is the second moment vector, $\beta_1, \beta_2 \in [0, 1)$ are the decay rates, with typical values as $\beta_1 = 0.9$, $\beta_2 = 0.999$ and ϵ is a small constant for numerical stability. The first and second moment vector are computed as:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \nabla_{\theta} \mathcal{L}(\theta_t), \quad (3)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) (\nabla_{\theta} \mathcal{L}(\theta_t))^2, \quad (4)$$

the bias is posteriorly recalculated as:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}, \quad (5)$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (6)$$

and then the parameters are updated:

$$\theta_{t+1} = \theta_t - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}, \quad (7)$$

where θ_t are the parameters at time step t , α is the learning rate (often set to 0.001 by default).

In CNNs, convolutional layers often have different gradient magnitudes compared to fully connected layers. Adam's adaptive learning rates help normalize this effect, ensuring stable and efficient learning across different parts of the network. (BENGIO; GOODFELLOW; COURVILLE, 2017). Additionally, Adam requires little hyperparameter tuning, making it especially popular in research and production pipelines.

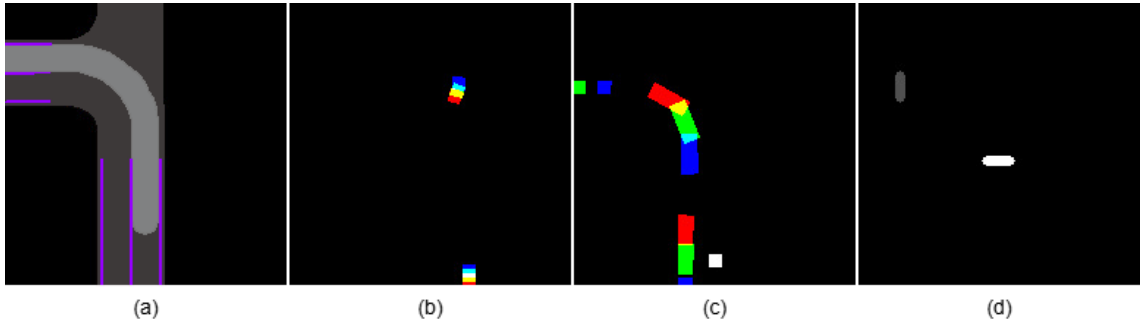
2.3 BIRD'S-EYE VIEW REPRESENTATION

The Bird's-Eye View (BEV) provides a top-down representation of the vehicle's surroundings, offering a structured and simplified depiction of the road environment (COUTO; ANTONELO, 2025; BANSAL; KRIZHEVSKY; OGALE, 2018). This representation enables the visualization of the vehicle's position, heading, and speed-denoted as p_t , θ_t , and s_t relative to the environment.

Figure 5 – **Camera images:** (a) is the camera image from the left side of the car, (b) is the frontal camera image and (c) is the right camera image. The cameras capture images from the simulator with its informations of road, lanes and obstacles.



Figure 6 – **BEV Representation:** (a) contain the desired route in light gray, drivable area in dark gray and lane boundaries in purple. Additionally, (b) represent the nearby pedestrians, (c) the nearby vehicle and (d) the traffic light states, the different time steps of (b) and (c) are represented by the colors red, green and blue, being red the most recent time step and blue the least, (b) (c) and (d) time step are encoded into separate BEV channels.



The BEV is centered on the vehicle, ensuring that its starting position remains fixed within the image frame. The environment is represented as a set of images of size $W \times H$ pixels, with a ground sampling resolution of ϕ meters per pixel. As the vehicle moves, the BEV dynamically updates, maintaining a consistent view of a fixed range of meters ahead. The BEV representation for the vehicle whose three frontal cameras are shown in Fig. 5 is given in Fig. 6.

The final BEV representation, combining spatial, pedestrians, vehicles and traffic signal information serves as input to the autonomous agent. This structured representation offers multiple advantages. It facilitates more efficient processing and interpretation. Moreover, mid-level visual inputs such as BEV have been shown to accelerate policy learning, enhance generalization, and improve overall performance in autonomous driving tasks (COUTO; ANTONELO, 2025). Recent works have also explored advanced techniques for generating BEV representations from multi-view camera inputs, further reinforcing their relevance and effectiveness in autonomous driving systems (SANTOS; ANTONELO; COUTO, 2025).

In the context of Implicit Behavioral Cloning (IBC), the BEV representation plays

a particularly important role by providing a compact and semantically structured state representation for learning the energy function over the action space. Since IBC relies on evaluating the compatibility between states and candidate actions through an energy-based formulation, a stable and spatially aligned representation such as BEV reduces perceptual ambiguity and facilitates the discrimination between expert actions and counterexamples. Recent studies on BEV generation from multi-view camera inputs further highlight the robustness and expressiveness of this representation, demonstrating its effectiveness in capturing scene geometry and dynamic elements in a unified top-down space (SANTOS; ANTONELO; COUTO, 2025). This makes BEV especially well-suited for IBC, as it supports multimodal action inference while improving generalization across different traffic scenarios.

2.4 BEHAVIOR CLONING

2.4.1 Introduction

Behavior Cloning (BC) is a foundational approach within Imitation Learning (IL), where the agent learns to replicate expert behavior by mimicking state-action pairs provided in a dataset (ANTONELO *et al.*, 2024). It is cast as a supervised learning problem in which the agent's policy is trained to minimize the difference between its predicted actions and those of the expert. In this work, BC is adopted as a baseline for performance comparison and also serves as the foundational strategy upon which our proposed method, Implicit Behavioral Cloning (IBC), is built.

When applied to self-driving tasks, BC aims to learn a policy that enables an autonomous vehicle to navigate effectively. It takes the BEV image and the current environmental state as inputs, and outputs the vehicle's acceleration and steering commands. To process the high-dimensional BEV images, a Convolutional Neural Network (CNN) is employed. The CNN captures spatial correlations and extracts hierarchical feature representations necessary for understanding lane topology, drivable areas, traffic signals, and the location of dynamic agents such as vehicles and pedestrians. Simultaneously, structured state information, such as speed and heading, is passed through a Multilayer Perceptron (MLP) that complements the visual input with motion context. The feature embeddings from both branches are then fused and passed through additional layers to regress the final action outputs.

"The loss function for BC is typically based on the difference between the actions predicted by the learned policy and the actions taken by the expert in the training data." (ANTONELO *et al.*, 2024).

2.4.2 Mathematical Formulation

To identify the variables, we consider: $\pi_\theta(s)$ as the policy parameterized by θ , which is responsible for mapping a state s to an action a . $\mathcal{D} = \{(s_i, a_i)\}_{i=1}^N$ is the dataset of N state-action pairs collected from the expert, where s_i is a state and a_i is the corresponding action taken by the expert.

BC's loss function commonly uses the mean squared error (MSE) between the predicted actions and the expert examples (ANTONELO *et al.*, 2024). If the actions are discrete, MSE is replaced by cross-entropy loss. However, the Beta distribution can be used to describe the policy, as it is more suitable for stochastic policies, also when variables are bounded within an interval, Beta distribution is suitable for modeling them.

In our work, the acceleration a_i^{acc} and steering a_i^{steer} actions are bounded within $[-1, 1]$. The set of actions is $a_i = (a_i^{\text{acc}}, a_i^{\text{steer}})$ and the policy $\pi_\theta(s_i) = (\alpha_i^{\text{acc}}, \beta_i^{\text{acc}}, \alpha_i^{\text{steer}}, \beta_i^{\text{steer}})$, where α and β are the shape parameters of the Beta distribution.

The probability density function of the Beta distribution (GELMAN *et al.*, 1995) is:

$$f(x; \alpha, \beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha, \beta)}, \quad (8)$$

where $B(\alpha, \beta)$ is the Beta function, a normalization constant to ensure that the total probability is 1. As the Beta distribution is only defined in the interval of $[0, 1]$, the actions are transformed from $[-1, 1]$ to $[0, 1]$ using the following equation:

$$a'_i = \frac{a_i + 1}{2} \quad (9)$$

The loss is the negative log-likelihood for the Beta distribution, which can be written as:

$$\mathcal{L}(\theta) = -\frac{1}{N} \sum_{i=1}^N \left[\log f\left(\frac{a_i^{\text{acc}} + 1}{2}; \alpha_i^{\text{acc}}, \beta_i^{\text{acc}}\right) + \log f\left(\frac{a_i^{\text{steer}} + 1}{2}; \alpha_i^{\text{steer}}, \beta_i^{\text{steer}}\right) \right] \quad (10)$$

This loss function accounts for the probabilistic nature of the Beta distribution, ensuring that the model learns to predict parameters (α, β) for the distribution of actions rather than the actions directly. This approach allows for modeling uncertainty and variability in the predicted actions (ANTONELO *et al.*, 2024).

This loss function leverages the probabilistic nature of the Beta distribution, training the model to predict the distribution parameters (α, β) instead of direct actions. This enables the modeling of uncertainty and variability in the predicted actions (ANTONELO *et al.*, 2024).

Despite its simplicity and effectiveness, BC suffers from the distributional shift problem: the policy is trained only on states seen in the expert data and may perform poorly when encountering out-of-distribution states at test time. This issue can compound over time, especially in sequential decision tasks, due to error accumulation.

Nevertheless, BC remains an essential method in IL and deep reinforcement learning pipelines due to its ease of implementation, compatibility with modern deep architectures, and scalability to large datasets. In the next sections, we build upon BC by introducing Energy-Based Models (EBM) and Implicit Behavioral Cloning (IBC), which improves generalization and robustness by modeling a latent distribution over actions through energy-based methods.

2.5 ENERGY-BASED MODELS

2.5.1 Introduction

EBMs are a class of models that define a scalar energy function $E_\theta(x)$, parameterized by neural networks or other differentiable functions, which associates a low energy to desirable or likely configurations of variables and high energy to unlikely ones (LECUN; CHOPRA, *et al.*, 2006). Unlike traditional probabilistic models that explicitly define normalized likelihoods, EBMs implicitly define a probability distribution through the Boltzmann distribution:

$$p(x) = \frac{e^{-E(x)}}{\int_{x \in X} e^{-E(x)}} = \frac{e^{-E(x)}}{Z}, \quad (11)$$

where $E(x)$ is the energy function of a given observation x and Z is the partition function that normalizes the distribution. However, in practice, computing Z is generally intractable due to the high dimensionality of the input space, which presents challenges for training.

If an input observation has lower energy than alternative observations, it is exponentially more likely under the distribution, but its absolute probability depends on the partition function. EBMs naturally allow multiple valid actions rather than forcing a single output. This enables the model to learn a more expressive representation of human-like driving behavior without averaging conflicting actions.

In Imitation Learning (IL), and more specifically in Implicit Behavioral Cloning (IBC) (FLORENCE *et al.*, 2022), EBMs are used to model the conditional distribution over actions a given a state s . This allows the learned policy $\pi_\theta(a|s) = \exp(-E_\theta(s, a))$ to flexibly represent multimodal behaviors, a crucial capability for applications such as autonomous driving, where multiple correct actions (e.g., braking or changing lanes) may be appropriate in the same state.

According to LeCun, Chopra, *et al.* (2006), training EBMs involves adjusting the energy function such that desired (correct) configurations of variables have lower energy than incorrect ones. This is typically achieved by minimizing a loss function that enforces an energy gap between positive (real) and negative (incorrect or generated) samples.

In deep EBMs, the energy function $E_\theta(x)$ is typically implemented using neural networks. For high-dimensional inputs such as images or BEV maps in autonomous driving, Convolutional Neural Networks (CNNs) or hybrid encoder-MLP architectures are employed to extract feature representations before computing energy values. This supports flexible and expressive modeling of complex state-action spaces.

2.5.2 Training

Training EBMs involves shaping the energy landscape such that expert or “positive” examples (low-energy) are separated from incorrect or synthesized “negative” examples (high-energy). This is typically done by minimizing a contrastive loss function that penalizes energy for positive samples and encourages high energy for negatives.

Since the partition function, which normalizes the energy-based probability distribution, is often intractable, direct maximum likelihood estimation is impractical. Instead, training often relies on approximate methods like contrastive divergence (CD) (HINTON, 2002), where negative samples are generated via a short Markov Chain Monte Carlo (MCMC) run from the current model, and difference of gradients (also known as energy contrastive estimation), which encourages the model to assign lower energy to correct examples while increasing the energy of incorrect ones. These methods allow EBMs to be trained efficiently without explicitly computing the partition function.

There are other methods that can be used, such as stochastic gradient Langevin dynamics (SGLD) (WELLING; TEH, 2011): A sampling technique that performs gradient-based updates with noise, allowing exploration of low-energy regions or score matching and denoising score matching (SONG; ERMON, 2021): Approaches that estimate the energy function gradient (score) directly, bypassing the partition function.

These techniques allow efficient training without requiring the exact computation of Z , which remains infeasible in most practical settings.

2.6 KERNEL DENSITY ESTIMATION

A Kernel Density Estimator (KDE) is a non-parametric method used to estimate the probability density function (PDF) of a random variable. It works by placing a kernel function centered at each data point in the sample. The overall density estimate at a point x is obtained by summing all these individual kernels and dividing by the total number of data points.

Mathematically, the KDE at a point x is defined as:

$$\hat{g}(x) = \frac{1}{Nh} \sum_{j=1}^N K\left(\frac{x - x_j}{h}\right), \quad (12)$$

where $\hat{g}(x)$ is the estimated density at point x , N is the number of data points, h is the bandwidth, x_j are the data points, and K is the kernel function.

The Gaussian kernel, commonly used in KDE, is a kernel function that assigns weights to data points based on their distance from the location where the density is being estimated. It is based on the standard normal distribution, which means that points closer to the target location contribute more to the estimate than those further away. The mathematical expression for the Gaussian kernel K is:

$$K(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}}, \quad (13)$$

where u is the standardized distance between the point where the density is being estimated and the data point, scaled by the bandwidth h .

In this work, the KDE is used to weigh the training samples in the BC and IBC losses in, respectively, (10) and (18). The resulting weighted BC loss function can be written as:

$$\mathcal{L}(\theta) = -\frac{1}{N} \sum_{i=1}^N w_i \left[\log f\left(\frac{a_i^{\text{ac}} + 1}{2}; \alpha_i^{\text{ac}}, \beta_i^{\text{ac}}\right) + \log f\left(\frac{a_i^{\text{steer}} + 1}{2}; \alpha_i^{\text{steer}}, \beta_i^{\text{steer}}\right) \right]. \quad (14)$$

As for the weighted IBC loss is as follows:

$$\mathcal{L}_{\text{InfoNCE}}(\theta) = \sum_{i=1}^N -w_i \log \left(\tilde{p}_{\theta}(y_i \mid x_i, \{\tilde{y}_i^j\}_{j=1}^{N_{\text{neg}}}) \right), \quad (15)$$

where: $w_i = \frac{1}{\hat{g}(a_i)}$; and $\hat{g}(a_i)$ is the kernel density estimation evaluated at the action a_i . The KDE is estimated for the actions in the training set. Then the weights w_i are calculated as the inverse of the KDE for each action.

In the training process, the weighted loss is not directly applied, in order to increase the frequency of less common actions of the dataset, at each iteration a minibatch is collected using a weighted sampling approach. The samples probability are defined by the weights to avoid data imbalance. This approach ensures that the loss function accounts for the density of actions in the training set, giving higher importance to less frequent actions. This improves the agent exploring capability, reducing the probability of the agent to get addicted to choosing the same path over time, in other words, improves multimodality behavior.

2.7 IMPLICIT BEHAVIORAL CLONING

Implicit Behavioral Cloning (IBC) is a learning method that integrates EBMs with IL to address the limitations of traditional (BC), particularly the issue of distributional shift (FLORENCE *et al.*, 2022).

While BC directly regresses an action $\hat{a} = F_{\theta}(x)$ from an observation x using supervised learning, IBC instead defines an energy function over the space of actions.

The policy does not directly output an action, but rather performs inference by optimization, selecting the action that minimizes the energy given the input observation:

$$\hat{a} = \operatorname{argmin} E_{\theta}(x, a), \quad (16)$$

This structure allows IBC to implicitly model multimodal action distributions, handle uncertainty, and better respect complex constraints in action spaces, which is especially useful in autonomous driving where diverse behaviors may be appropriate for the same state. See Figure 2 for a visual comparison between explicit BC regression and implicit energy-based action inference.

2.7.1 Training

The dataset of samples is $\{x_i, y_i\}$ and the regression bound are $y_{min}, y_{max} \in \mathbb{R}^m$. To train the model and achieve a good result, it is first needed to generate a set of negative counter-examples, which are described as $\{\tilde{y}_i^j\}_{j=1}^{N_{neg}}$ for each sample x_i and applying an InfoNCE-style loss function (FLORENCE *et al.*, 2022).

The loss can be defined as the negative log likelihood of the probability distribution described as follows:

$$p_{\theta}(y|x, \{\tilde{y}_i^j\}_{j=1}^{N_{neg}}) = \frac{e^{-E_{\theta}(x, y)}}{e^{-E_{\theta}(x, y)} + \sum_{j=1}^{N_{neg}} e^{-E_{\theta}(x, \tilde{y}_i^j)}}, \quad (17)$$

where to estimate the denominator we use the counter examples. Then, the loss can be determined as:

$$\mathcal{L}_{\text{InfoNCE}} = \sum_{i=1}^N -\log p_{\theta}(y_i|x_i, \{\tilde{y}_i^j\}_{j=1}^{N_{neg}}), \quad (18)$$

This formulation allows the model to avoid directly regressing to the mean of multimodal action distributions, naturally express stochastic or ambiguous behaviors and handle complex, high-dimensional action spaces using structured sampling or learned proposals.

2.7.2 Advantages of IBC

IBC presents several advantages over BC: improved generalization, as it avoids overfitting to the mean of conflicting demonstrations a common issue in BC; the ability to represent uncertainty in ambiguous scenarios through energy functions rather than direct outputs; and, due to the contrastive nature of the InfoNCE objective, the capacity to learn robust distinctions between valid and invalid actions with relatively few samples. However, reducing the total number of training samples can limit the diversity and quality of the counterexamples, even if their number (N_{neg}) remains constant.

2.7.3 Inference using Derivative-Free Optimization (DFO)

With the model E_θ trained, implicit inference is performed with stochastic optimization. The optimal action in IBC is inferred by:

$$\hat{a} = \arg \min_a E_\theta(s, a). \quad (19)$$

DFO is a sampling-based approach that selects actions by minimizing $E_\theta(s, a)$ without requiring gradients. This makes it particularly useful for optimization over non-smooth or multimodal distributions. Common variants include evolutionary strategies and stochastic search methods. The process consists of:

1. **Sampling:** Generate candidate actions $\{\tilde{a}_i\}$ uniformly.
2. **Evaluation:** Compute $E_\theta(s, \tilde{a}_i)$ for each sample.
3. **Selection:** Choose the action with the lowest energy.

This approach enables robust action selection in complex scenarios without explicit gradient computation, and is described also in Florence *et al.* (2022). In this context, implicit models can approximate discontinuities and multi-valued functions, where a probabilistic decision happens between a set of optimal values.

The procedure followed by the Derivative-Free Optimization (DFO) method is formally detailed in Algorithm 1, which outlines the sampling-based approach used to iteratively refine candidate solutions in the absence of gradient information.

Algorithm 1: Derivative-Free Optimizer

Result: $\hat{y}$

Initialize: $\{\hat{y}_i\}_{i=1}^{N_{\text{samples}}} \sim \mathcal{U}(y_{\min}, y_{\max}), \sigma = \sigma_{\text{init}}$

for iter in $1, 2, \dots, N_{\text{iters}}$ **do**

$\{E_i\}_{i=1}^{N_{\text{samples}}} = \{E_\theta(x, \hat{y}_i)\}$ {compute energies}

$\{p_i\}_{i=1}^{N_{\text{samples}}} = \left\{ \frac{e^{-E_i}}{\sum_{j=1}^{N_{\text{samples}}} e^{-E_j}} \right\}$ {softmax}

if iter $< N_{\text{iters}}$ **then**

$\{\hat{y}_i\}_{i=1}^{N_{\text{samples}}} \sim \text{Multinomial}(N_{\text{samples}}, \{p_i\}_{i=1}^{N_{\text{samples}}}, \{\hat{y}_i\}_{i=1}^{N_{\text{samples}}})$

$\{\hat{y}_i\}_{i=1}^{N_{\text{samples}}} \leftarrow \{\hat{y}_i\}_{i=1}^{N_{\text{samples}}} + \mathcal{N}(0, \sigma)$ {add noise}

$\{\hat{y}_i\}_{i=1}^{N_{\text{samples}}} \leftarrow \text{clip}(\{\hat{y}_i\}_{i=1}^{N_{\text{samples}}}, y_{\min}, y_{\max})$ {clip to bounds}

$\sigma \leftarrow K\sigma$ {shrink sampling scale}

end if

end for

$\hat{y} = \arg \max(\{p_i\}, \{\hat{y}_i\})$

The algorithm begins by sampling a set of candidate solutions (denoted by $\hat{y}_i$) from a uniform distribution between $y_{\min}$ and $y_{\max}$. These represent possible values of the parameters (or actions, in some machine learning contexts) that will be evaluated.

$N_{samples}$ is the number of samples (i.e., candidate solutions), and σ_{init} is the initial value for the noise scale, which controls the amount of randomness introduced during the optimization.

The main loop is initialized with N_{iters} . The algorithm then calculates an energy value E_i for each sample. This could represent a cost or loss, or some other evaluation of the solution based on the objective function. The energy is typically calculated via some function $E_\theta(x, \hat{y}_i)$ that evaluates each candidate solution.

After obtaining the energies, the algorithm computes a probability distribution over the solutions using a softmax function, described in equation 20.

$$p_i = \frac{e^{-E_i}}{\sum_{j=1}^{N_{samples}} e^{-E_j}} \quad (20)$$

The probability p_i of selecting a solution is based on its energy. Lower energy values (better solutions) will have higher probabilities. The softmax normalizes the energies to give a valid probability distribution. The samples are then updated. If the current iteration is less than the total number of iterations N_{iters} , the algorithm proceeds to update the candidate solutions. Each sample $\hat{y}_i$ is updated according to the multinomial distribution using the probabilities p_i . This step biases the selection toward the solutions with lower energies (better solutions).

The algorithm introduces some random noise to each sample. This is done by adding Gaussian noise $\mathcal{N}(0, \sigma)$ to each solution, where σ is the current noise scale. This introduces exploration into the optimization process, preventing the algorithm from prematurely converging to a local minimum.

After adding noise, the solutions are clipped to ensure they remain within the bounds defined by y_{min} and y_{max} . This step ensures that the solutions stay within feasible regions.

The noise scale σ is then shrunk by a factor K . This reduces the randomness over time, helping the algorithm refine the search in later iterations by focusing on more promising regions of the solution space.

After completing all iterations, the algorithm selects the best solution based on the highest probability p_i . This solution corresponds to the maximum likelihood solution among the candidates. The final result is $\hat{y} = \arg \max(\{p_i\}, \{\hat{y}_i\})$, which selects the solution with the highest associated probability.

2.7.4 IBC Architecture

The proposed architecture is shown in Figure 7, it is an Energy-Based Model (EBM) designed for Implicit Behavioral Cloning (IBC) in autonomous driving. Its objective is to learn an energy function $E(x, y)$ that assigns low energy values to expert-like actions y given a driving state x , while assigning higher energy values to suboptimal

actions. By explicitly conditioning the energy function on the action space, the model is able to represent multimodal decision-making and avoid the action-averaging behavior commonly observed in standard Behavioral Cloning approaches.

The perception module receives as input a Bird's-Eye View (BEV) representation of the environment with dimensions $4 \times 192 \times 192$. The four channels encode information about the driving scene, such as road layout, lane markings, obstacles, and dynamic agents. This input is processed through a convolutional backbone composed of successive convolutional layers with ReLU activations. The network begins with three 5×5 convolutional layers with stride 2, producing feature maps with 8, 16, and 32 channels, respectively, while progressively reducing spatial resolution. This is followed by a 3×3 convolution with 64 channels and stride 2, further compressing the spatial representation while increasing semantic abstraction.

To refine the extracted spatial features, an additional 3×3 convolutional layer with 128 channels and stride 2 is applied, followed by another 3×3 convolution with 256 channels and stride 1. The resulting feature map is flattened, yielding a 1024-dimensional vector that represents a compact encoding of the scene context. This BEV encoder captures both geometric and semantic information relevant for driving decisions, such as intersections, road boundaries, and surrounding traffic participants.

In parallel to the perception stream, the architecture processes a low-dimensional vehicle state vector composed of six elements: the previous throttle, steering, brake, and gear commands, as well as the longitudinal and lateral velocities (v_x, v_y) . This vector is passed through two fully connected layers with 256 units and ReLU activations, producing a latent embedding that captures short-term vehicle dynamics and control history. The encoded vehicle state is then concatenated with the BEV feature vector, forming a unified representation of the environment and the ego-vehicle state.

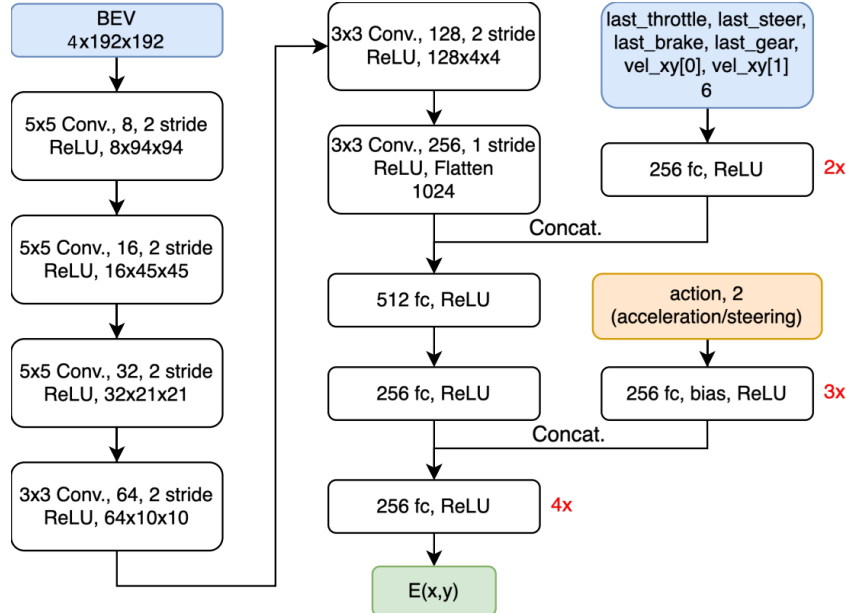
The combined state representation is further processed by two fully connected layers with 512 and 256 units, respectively, both using ReLU activations. This stage produces a joint latent state embedding that serves as the conditioning input for the action-dependent energy evaluation.

To incorporate action conditioning, the candidate action vector, composed of acceleration and steering commands, is processed by a dedicated action encoder consisting of three fully connected layers with 256 units, bias terms, and ReLU activations. This action embedding is then concatenated with the joint state representation, allowing the network to evaluate the compatibility between a given state and a specific action.

Finally, the concatenated state–action representation is passed through a sequence of four fully connected layers with 256 units and ReLU activations. The network outputs a single scalar value corresponding to the energy $E(x, y)$. Lower energy values indicate actions that are more consistent with expert demonstrations, while higher values correspond to less desirable behaviors. During inference, actions are selected

by minimizing this learned energy function, enabling the model to naturally express multimodal driving behaviors in ambiguous scenarios such as intersections.

Figure 7 – **IBC architecture**: Architecture of the energy-based CNN policy. Spatial features extracted from the input image are combined with vehicle control states (previous throttle, steering, brake, gear, and velocity) and a proposed action (acceleration and steering) to produce the corresponding energy value.



Furthermore, to improve the agent's capability to respect the traffic lights, we use Kernel Density Estimation (KDE) to estimate weights for the training samples in our loss function in (18), as done and described in (ANTONELO *et al.*, 2024).

3 METHODOLOGY

In this chapter, it is presented the methodological aspects of this work. Describing the research done, the software used, and the hardware needed to gather the results, with the aim to validate the hypotheses.

3.1 METHODOLOGICAL PROCEDURE

The methodological procedure adopted in this research is structured in distinct phases, each contributing to the development, implementation, and evaluation of an autonomous driving agent based on IBC technique. The approach integrates both theoretical investigation and practical experimentation.

3.1.1 Courses

Initially, a comprehensive analysis of postgraduate course subjects was undertaken to consolidate core theoretical concepts in machine learning. This enabled the identification and selection of appropriate methodologies applicable to imitation learning and autonomous driving. In addition, a bibliographic review was conducted. This review encompassed key topics such as BC, IBC, derivative-free optimization strategies, and energy-based models techniques.

To further expand technical proficiency, supplementary external courses were undertaken. These included a course in machine learning with a focus on imitation learning paradigms, as well as programming courses emphasizing proficiency in Linux-based systems and Python programming language. These skills were essential for the development and debugging of the autonomous driving algorithms used throughout this work.

3.1.2 CARLA Simulator

The experimental phase of this research was conducted using the CARLA simulator, a high-fidelity, open-source simulation platform specifically designed for autonomous driving research and development. CARLA provides realistic and configurable urban environments, enabling the controlled evaluation of learning-based driving policies under conditions that closely resemble real-world scenarios. The simulator is built on the Unreal Engine, which allows for photorealistic rendering, accurate vehicle dynamics, and detailed modeling of road infrastructure.

CARLA supports a wide range of functionalities essential for autonomous driving experiments, including predefined and user-defined navigation routes, dynamic traffic agents such as vehicles and pedestrians, traffic lights and road signage, as well as configurable weather and lighting conditions (e.g., rain, fog, day–night cycles). These

capabilities enable the systematic testing of agent behavior in complex and multimodal scenarios, such as intersections, lane merges, and urban traffic interactions, while maintaining full control over environmental variables and experimental reproducibility.

In addition, CARLA provides multiple sensor modalities commonly used in autonomous vehicles, such as RGB cameras, depth sensors, semantic segmentation, LiDAR, and radar. In this work, the agent operates using a Bird's-Eye View (BEV) representation derived from the simulator state, which abstracts the scene geometry and surrounding traffic participants, facilitating learning while preserving critical spatial information.

The implementation of the driving agent and its integration with the CARLA simulator were carried out using Python, leveraging CARLA's Python API for environment control, sensor management, and data acquisition. Development was conducted in Visual Studio Code (VSCode), which provided an efficient environment for code management and experimentation. Python was chosen due to its extensive ecosystem of machine learning libraries and its seamless compatibility with CARLA, enabling rapid prototyping, training, and evaluation of the proposed algorithms.

An expert dataset was generated to train the IBC agent, simulating expert driving behavior across various scenarios, which is not available online. The training process involved repeated experimentation, during which the model parameters were iteratively adjusted based on performance metrics. Scenarios were gradually scaled in complexity, from simple fixed routes to dynamic urban environments with multiple moving agents and obstacles, to rigorously evaluate the robustness and adaptability of the trained policy.

Performance evaluation considered key indicators such as route completion rate, number of collisions, respect to traffic lights, and the agent's ability to make diverse decisions in multimodal contexts.

3.2 RESOURCES

The development and execution of this research relied on a combination of hardware, software, institutional, and financial resources, as detailed below:

1. **Desktop Computer:** Equipped with an Intel Core i5-10400F processor, NVIDIA GeForce RTX 2080 Super graphics card, 16 GB of RAM, and 1 TB of SSD storage. This machine was primarily used for training neural network models and running simulation scenarios in the CARLA environment.
2. **Notebook Computer:** Configured with an Intel Core i7-11370H processor, NVIDIA GeForce RTX 3060 graphics card, 16 GB of RAM, and 1 TB of SSD storage. This device supported development tasks, code debugging, and scenario visualization during mobility periods.

3. **UFSC High-Performance Computing Server:** Access to the university's internal server infrastructure was granted, providing high processing power for accelerated model training and parallel experiments. Configured with an Intel Xeon 8 cores processor, GeForce RTX 3080 graphics card with 10Gb, and 64 Gb of RAM.
4. **Source Code Repository:** Utilization of the IBC implementation made publicly available by the original authors. This served as the foundation for adaptation, experimentation, and performance evaluation. Available in: <https://github.com/google-research/ibc>
5. **Financial Support:** The research was supported by a master's scholarship provided by the Fundação de Amparo à Pesquisa e Inovação do Estado de Santa Catarina (FAPESC), in the process 48/2021, which enabled full-time dedication to the project and acquisition of complementary resources.

4 RELATED WORK

This chapter reviews the main imitation learning paradigms that form the foundation of this work, with a particular focus on their applicability to autonomous driving and their ability to handle multimodal decision-making. The discussion begins with traditional Behavior Cloning (BC) approaches and their limitations in complex driving scenarios, followed by implicit formulations based on Energy-Based Models (EBMs) and Implicit Behavioral Cloning (IBC). These methods are followed by generative diffusion-based policies and adversarial approaches such as Generative Adversarial Imitation Learning (GAIL), highlighting their respective strengths and practical challenges. Through this review, it is contextualized the methodological choices adopted in this dissertation and motivate the proposed Data-Augmented IBC (DA-IBC) framework as a balanced solution that targets multimodality, robustness, and real-time feasibility.

4.1 BEHAVIOR CLONING

Behavioral Cloning (BC) is one of the earliest and most widely studied imitation learning approaches for autonomous driving. In BC, the driving policy is learned via supervised learning by directly mapping observed states to expert actions using datasets of state–action pairs collected from human demonstrations. By avoiding explicit reward modeling and online environment interaction, BC offers a conceptually simple and data-efficient framework when high-quality expert demonstrations are available.

Early applications of BC date back to the work of Pomerleau (1989), who introduced the ALVINN system, demonstrating that a neural network could learn to map camera images directly to steering commands from human driving data. This work laid the foundation for end-to-end learning in autonomous vehicles and remains a cornerstone of the behavior cloning literature. With the resurgence of deep learning, BC gained renewed attention through large-scale implementations such as the work of Bojarski *et al.* (2016), who demonstrated end-to-end driving using deep convolutional neural networks trained on real-world camera data. While their approach showed promising generalization under controlled conditions, it also exposed BC’s sensitivity to dataset bias and distributional shift.

Subsequent studies explored BC in simulated urban environments to better analyze its limitations under controlled yet complex scenarios. Codevilla *et al.* (2018) introduced Conditional Imitation Learning (CIL), in which the policy is conditioned on high-level navigation commands such as *turn left*, *go straight*, or *turn right*. Using the CARLA simulator, the authors proposed the NoCrash benchmark to evaluate driving performance under dynamic and high-traffic conditions. Although BC-based methods achieved strong results in structured settings, their performance degraded in more complex environments due to limited generalization, overfitting, and compounding errors.

From a theoretical standpoint, the fundamental limitations of BC in sequential decision-making were formalized by Ross *et al.* (2011), who identified the issue of *covariate shift*. Since BC is trained exclusively on expert state distributions, small prediction errors during deployment can drive the agent into unseen states where the learned policy is unreliable, causing errors to accumulate over time. While approaches such as Dataset Aggregation (DAgger) alleviate this issue by incorporating on-policy data, they require interactive expert supervision, which is often impractical in autonomous driving contexts.

More recent work has focused on improving BC through alternative state representations and data-efficient training strategies. For example, Yildirim *et al.* (2024) demonstrated the feasibility of BC for steering angle prediction using CNNs trained on visual inputs, highlighting its ability to reproduce human-like driving behavior in controlled environments. Similarly, Antonelo *et al.* (2024) investigated BC using Bird's-Eye View (BEV) representations in simulated urban settings, emphasizing the importance of structured mid-level representations, data augmentation, and domain adaptation to improve robustness when training from limited demonstrations. Despite these advances, BC remains fundamentally constrained by its deterministic policy formulation.

A central limitation of standard BC is its inability to represent multimodal action distributions. In scenarios where multiple valid driving behaviors exist—such as choosing different maneuvers at an intersection BC tends to regress toward the mean action, resulting in unrealistic or suboptimal behavior. These limitations motivate the exploration of alternative policy representations capable of modeling multimodal decision-making. In this context, Implicit Behavioral Cloning (IBC) introduces an energy-based formulation that models the compatibility between states and actions, allowing the policy to represent multiple valid behaviors without explicit reward signals. Building upon this idea, Data-Augmented IBC (DA-IBC) further refines the learning process by incorporating perturbations of expert actions, improving generalization and training stability. These approaches aim to overcome the structural limitations of traditional BC and provide a more scalable and robust solution for autonomous driving in complex and ambiguous environments.

4.2 IMPLICIT BEHAVIOR CLONING AND ENERGY-BASED MODELS

While standard Behavior Cloning (BC) learns an explicit deterministic mapping from states to actions via supervised regression, Implicit Behavioral Cloning (IBC) reformulates imitation learning by modeling the compatibility between states and actions through an energy function. In BC, the policy directly outputs a single action, which leads to action averaging and brittle behavior in multimodal scenarios where multiple valid actions exist. In contrast, IBC defines a conditional energy landscape over the action space, assigning low energy to expert-like actions and higher energy to suboptimal

alternatives, thereby representing a full implicit action distribution rather than a single point estimate. Action selection in IBC is performed by minimizing the learned energy function at inference time, enabling the recovery of diverse behaviors consistent with expert demonstrations without requiring explicit likelihood modeling or reward design. This energy-based formulation allows IBC to naturally capture multimodal decision-making and mitigates the mode-collapse and covariate-shift issues commonly observed in standard BC, making it particularly suitable for complex autonomous driving scenarios with inherent ambiguity.

Balesni, Tampuu, and Matiisen (2023) investigated the use of Energy-Based Models (EBMs) in conjunction with Implicit Behavioral Cloning (IBC) to control the steering of autonomous vehicles. While their method employs a CNN-based encoder and build the EBM using several supervised loss formulations, tailored to discrete or low-dimensional action outputs, the approach proposed in this dissertation adopts a contrastive energy formulation inspired by InfoNCE (FLORENCE *et al.*, 2022), which interprets imitation learning as an energy minimization process over implicit action distributions. This design removes the need to impose parametric assumptions over the action space and improves the model’s ability to represent multimodal and continuous behaviors an essential property for complex driving scenarios.

It is important to notice that, the system of Balesni, Tampuu, and Matiisen (2023) controls only a single degree of freedom, steering, while velocity and positional updates are directly inherited from expert demonstrations. This simplification facilitates training stability, however, it limits the model’s capacity to adapt to variations in longitudinal dynamics and environmental uncertainty. In contrast, the method developed in this work learns both steering and acceleration, enabling end-to-end policy learning that better represent a real driving agent.

Performance-wise, Balesni, Tampuu, and Matiisen (2023) report that EBM-based policies can approximately match, but not overcome, the results of standard BC in steering tasks. Their EBM agents achieved comparable success rates on benchmark routes but exhibited higher lateral jerk, indicative of abrupt steering transitions that compromise smoothness and comfort. Although the authors implemented two regularization techniques to reduce the jerk, they did not produced significant performance gains relative to BC, suggesting that additional structure or regularization is required when applying EBMs to continuous control.

Beyond this, other researchers have expanded IBC beyond simple control tasks. Z. Zhang *et al.* (2024) combined IBC with Dynamic Movement Primitives to enhance robot motion planning, demonstrating improved transferability and sample efficiency. Similarly, Florence *et al.* (2022) validated the flexibility of IBC across diverse manipulation and locomotion benchmarks, showing that implicit energy formulations enable robust multimodal imitation without explicit likelihood modeling. This thesis aim to use

IBC in multimodal autonomous driving, proposing a data-augmented and contrastive energy-based framework (DA-IBC) that learns smooth, high-dimensional control policies and improves generalization to unseen driving contexts.

4.3 DIFFUSION MODELS

Diffusion models have recently gained popularity in imitation learning (IM) and planning for autonomous driving because they naturally represent multimodal action and trajectory distributions (LIAO *et al.*, 2024). By learning a denoising process, these models can generate diverse, plausible trajectories that reflect the different valid choices a human driver might take in a given scene. Studies in the offline behavior cloning show that diffusion policies can produce smoother and more varied behaviors than mean-regression BC, improving coverage of rare modes in driving datasets (MACHADO; ANTONELLO, 2025; LIAO *et al.*, 2024).

A key practical limitation of vanilla diffusion policies is their iterative denoising procedure, which raises inference latency and complicates integration into closed-loop control (LIAO *et al.*, 2024). To address this problem, the authors introduce architectural and algorithmic accelerations, DiffusionDrive proposes a truncated diffusion design with multi-mode priors and a decoder that reduces denoising steps, while maintaining high diversity and sample quality (LIAO *et al.*, 2024). Complementary evaluations in offline settings similarly highlight the trade-off between sampling cost and closed-loop performance, emphasizing the need for careful scheduling or truncation when using diffusion policies in latency-sensitive driving loops (MACHADO; ANTONELLO, 2025).

Furthermore, diffusion planners have flexible guidance mechanisms that bias sampling towards safety, comfort, or task-specific constraints (ZHENG *et al.*, 2025). This architecture integrates conditional diffusion with classifier guidance and transformer-based context fusion, demonstrating that guidance gradients can steer sampled trajectories toward desirable objectives, while preserving multimodality. However, this guidance introduces additional tuning complexity and increase runtime that must be managed for robust closed-loop behavior (ZHENG *et al.*, 2025; LIAO *et al.*, 2024).

Recent studies, such as Machado and Antonelo (2025), explore important trade-offs for diffusion models: (1) sampling-based diversity vs. inference latency, (2) flexibility via guidance vs. the need for careful conditioning design, and (3) sampling stability vs. closed-loop smoothness. Machado and Antonelo (2025) study highlights these trade-offs in realistic driving benchmarks, showing that diffusion models can improve diversity and reduce some failure modes of BC but require fine tuning to remain stable in closed-loop evaluations. Likewise, truncated and guided diffusion variants show promising paths to mitigate these issues but do not completely eliminate the need for temporal regularization and smoothing (LIAO *et al.*, 2024; ZHENG *et al.*, 2025).

DA-IBC targets the same practical multimodal capability and real-time viabil-

ity, but approaches them from an EBM perspective rather than by sampling from a generative diffusion chain. Where diffusion methods represent multimodality explicitly through multi-step sampling and optional guidance, DA-IBC learns a contrastive, InfoNCE-inspired energy landscape that admits efficient energy minimization for inference and naturally supports temporal regularizers and data augmentation for smoother outputs (FLORENCE *et al.*, 2022). Thus, DA-IBC draws conceptual lessons from diffusion research (the value of rich multimodal priors and the benefits of guidance/truncation) while prioritizing inference efficiency and temporal consistency for control complementary design choices motivated by the empirical trade-offs reported in Machado and Antonelo (2025), Zheng *et al.* (2025) and Liao *et al.* (2024).

4.4 GENERATIVE ADVERSARIAL IMITATION LEARNING (GAIL)

Generative Adversarial Imitation Learning (GAIL) was introduced by Ho and Ermon (2016), which combines principles from Inverse Reinforcement Learning (IRL) and Generative Adversarial Networks (GANs). Unlike traditional IRL methods that aim to explicitly recover a reward function from expert demonstrations, GAIL bypasses reward modeling entirely by optimizing the policy to imitate expert behavior through adversarial learning (HO; ERMON, 2016).

A policy is trained to generate trajectories that a discriminator cannot distinguish from those produced by an expert (LI; SONG; ERMON, 2017). The discriminator serves as a learned reward signal, guiding the policy to match the distribution of expert behavior (COUTO; ANTONELLO, 2025). This adversarial process enables the policy to generalize beyond supervised behavior cloning and to recover complex expert strategies without manual rewards.

GAIL offers several advantages over classical BC methods, particularly in terms of robustness to compounding errors and improved generalization to unseen states (COUTO; ANTONELLO, 2025). However, its reliance on policy reinforcement learning introduces significant drawbacks in terms of sample inefficiency and training instability. These challenges are particularly pronounced in high-dimensional or safety-critical domains such as autonomous driving, where data collection and environment interaction can be costly or impractical.

Despite these limitations, GAIL has become a foundational method in IL research and has inspired a wide range of extensions, including approaches that improve GAIL's training process, reduce sample complexity, or adapt the framework for vision-based control and continuous action spaces. Its influence remains significant in the development of adversarial and reward-free learning paradigms, including methods for learning from raw demonstrations in autonomous driving contexts.

While GAIL represents a significant advancement in IL, its dependence on adversarial training and reinforcement-based optimization contrasts with the more direct,

EBM formulation explored in IBC. IBC leverages an energy function to represent the compatibility between states and actions, allowing it to learn multimodal behaviors without requiring adversarial objectives or explicit reward signals. Consequently, approaches such as DA-IBC build upon this foundation to improve generalization and stability while maintaining sample efficiency in autonomous driving tasks.

4.5 COMPARISON OF METHODS

Table 1 provides the comparison of imitation learning methods commonly used in autonomous driving, focusing on their policy formulation, ability to represent multimodal behavior, inference cost, and main challenges. The comparison highlights structural differences rather than implementation details or empirical performance. In particular, the table contrasts explicit regression-based policies, such as Behavior Cloning, with implicit formulations based on energy models, generative diffusion processes, and adversarial reinforcement learning. This abstraction highlights how different policy representations impact multimodality, deployment efficiency, and failure modes in closed-loop driving, and serves to contextualize the methodological choices adopted in this work.

Table 1 – Comparison of major imitation learning paradigms for autonomous driving.

Method	Policy Formulation	Multimodality	Inference Cost	Challenges
BC	Explicit regression	Low	Low	Action averaging Covariate shift
IBC	Implicit (EBM)	High	Medium	Energy optimization at inference
Diffusion	Generative policy	Very High	High	Sampling latency Deployment complexity
GAIL	Explicit policy (RL)	Implicit	Low	Sample inefficiency Training instability

Behavioral Cloning (BC) is presented in the first row of 1, it relies on an explicit policy formulation based on supervised regression, directly mapping observations to actions. While this approach benefits from a low inference cost and straightforward training, its ability to represent multimodality is inherently limited. As shown in the table, BC tends to average actions across demonstrations, which often results in suboptimal behavior in ambiguous scenarios. This limitation is further compounded by covariate shift, where small prediction errors accumulate over time, leading the agent into unseen states and degraded performance during long-horizon rollouts.

Implicit Behavioral Cloning (IBC) addresses several shortcomings of BC by modeling the policy implicitly through an Energy-Based Model (EBM). Instead of predicting actions directly, IBC assigns lower energy to expert-like actions, enabling the represen-

tation of multiple valid behaviors for a given state. As indicated in Table 1, this implicit formulation allows IBC to highly adapt to multimodal scenarios, which is particularly beneficial in scenarios with multiple feasible maneuvers. However, this expressiveness comes at the cost of increased inference complexity, as action selection requires solving an energy optimization problem at test time, resulting in a medium inference cost.

Diffusion-based policies further extend the ability to model complex and highly multimodal action distributions by employing generative modeling techniques. These methods achieve very high multimodality, making them well-suited for capturing the full diversity of expert demonstrations in autonomous driving tasks. Nevertheless, their inference cost is significantly higher due to the iterative sampling process required to generate actions. As highlighted in the table, this sampling latency and the resulting deployment complexity pose practical challenges for real-time control, especially in safety-critical applications.

Generative Adversarial Imitation Learning (GAIL) represents a distinct paradigm by framing imitation learning as an online reinforcement learning problem guided by adversarial training. Although the policy itself is explicitly parameterized and incurs a low inference cost, multimodality is captured implicitly through the learned reward signal rather than directly through the policy distribution. Despite this advantage, GAIL suffers from well-known challenges, including sample inefficiency and training instability, which can limit its applicability in large-scale or computationally constrained settings.

The methods reviewed in this chapter reveal a clear progression in imitation learning for autonomous driving, from simple regression-based policies toward more expressive formulations capable of representing multimodal behavior. While BC remains attractive for its simplicity and low inference cost, its inability to cope with ambiguity and distribution shift limits its effectiveness in realistic urban environments. More expressive approaches, such as IBC, diffusion-based policies, and GAIL, address these limitations through implicit modeling, generative sampling, or adversarial learning, but introduce new challenges related to inference latency, training stability, or sample efficiency. These trade-offs directly inform the experimental design and evaluation presented in the following chapters. The following sections, experiments and results, investigate how the proposed DA-IBC framework positions itself within this landscape, demonstrating its ability to retain multimodal expressiveness while achieving stable closed-loop performance and competitive efficiency in realistic autonomous driving scenarios.

5 EXPERIMENTS AND RESULTS

5.1 EXPERT AND AGENT

5.1.1 Expert Dataset

The expert demonstration dataset was generated using CARLA simulator, a high-fidelity open-source driving simulator designed for autonomous driving research. Specifically, 10 predetermined routes from *town01* environment were selected gathered from CARLA Leaderboard evaluation platform¹ (Fig. 8). These routes were chosen to cover a diverse set of driving scenarios, including urban intersections, straight segments, and curved roads, thereby ensuring that the collected dataset would reflect a broad spectrum of navigation behaviors required in real-world urban driving.

The expert agent used to generate these demonstrations consists of a deterministic rule-based navigation policy, which follows a precomputed trajectory generated from a topological planner. This trajectory defines a sequence of waypoints corresponding to the planned route. For low-level control, the agent employs a classical Proportional-Integral-Derivative (PID) controller to regulate both throttle-brake commands (for longitudinal control) and steering (for lateral control), based on the current vehicle pose and the desired trajectory. This expert setup is commonly used in CARLA benchmarks and provides reliable and consistent driving behavior that is free from noise or learning-induced suboptimality, making it suitable for imitation learning experiments (CHEN *et al.*, 2019).

This expert dataset serves as the guideline for the agent, providing state-action pairs that reflect expert driving decisions under a variety of environmental conditions, including traffic, dynamic objects, and traffic light constraints, as configured within the simulator environment.

The *town01* environment used in this study includes one of the predefined routes utilized by the expert agent during data collection. This particular route, is shown in Fig. 8, spans a total length of approximately 740 meters and encompasses various urban driving challenges, including intersections, turns, and dynamic traffic elements. Also, in Fig. 9 to evaluate the agent learning and behavior it is used *Town02* to avoid a previously known environment by the agent.

Each observation in the dataset is represented as a multi-channel spatial input comprising twelve distinct channels, designed to encode both static and dynamic semantic information relevant to the driving task. Specifically, the first three channels encode static features of the environment: the road channel, which captures the drivable surface; the lane channel, which defines lane boundaries and lane center-lines; and the route channel, which encodes the expert’s planned trajectory as a dense sequence

¹ <https://leaderboard.carla.org/>

Figure 8 – Top-view image of CARLA's Town01 the dots are waypoints that define the route, the route start in the light-yellow and end at the red dot.



Figure 9 – Top-view image of CARLA's Town02 the dots are waypoints that define the route, the route start in the light-yellow and end at the red dot.



of way-points projected onto the bird's-eye view.

Channels 4 to 6 represent the temporal information of the pedestrians in time steps, as for channels 7 to 9 gather the information of the vehicles in the same time steps and the remaining three channels are used to encode temporal information about traffic light over three time steps, allowing the agent to observe changes in the dynamic obstacles and make anticipatory decisions accordingly. These channels are particularly relevant for tasks such as stopping, yielding, or proceeding through intersections under realistic constraints.

In the context of this work, in some experiments the route channel is intentionally excluded during training and inference. This decision is motivated by the goal of evaluating the agent's ability to learn multimodal behaviors in the absence of explicit goal conditioning. By removing the route input, the learning task becomes inherently ambiguous and requires the agent to infer plausible future actions based on its understanding of the road and traffic context. This design choice challenges the policy to represent uncertainty over possible expert behaviors, thereby aligning with the objectives of multimodal imitation learning frameworks such as Implicit Behavioral Cloning.

The expert set consists of ten recorded trajectories at 10 Hz, generating 10 observation-action pairs per second. The shortest trajectory contains 1480 samples (2.5 minutes), while the average trajectory has 2129 samples (3.5 minutes). In total, the dataset includes 21,287 labeled samples, covering approximately 36 minutes or 8 km of driving.

The learning rate is adjusted using Adam optimizer, with step size of $lr = 1e^{-5}$ for a maximum of 600 epochs, the minibatch size is $m = 128$. In EBM, to each sample we generate $N = 1024$ counterexamples. For DFO inference it is used $N_{samples} = 16.384$ samples, the standard noise $\sigma = 0, 5$, $N_{iters} = 5$ and $K = 0.5$. The kernel bandwidth is $h = 0.2$.

5.1.2 Image input

The agent receives as input a twelve-channel 192×192 bird's-eye view (BEV) image generated by a CARLA module with access to the city map. As detailed in Section 5.1.1, this BEV representation comprises three channels for the road, route, and lane layout; three channels encoding pedestrian positions across time steps; three channels representing vehicle positions; and three channels corresponding to traffic light states, all captured at time steps -16, -9, and -1. Depending on the experiment, specific channel groups may be included or excluded. In addition to the BEV image, the agent also receives the vehicle's current speed and the previous control actions (acceleration and steering).

5.1.3 Action Representation

The vehicle in the CARLA simulator has three actuators: steering $[-1, 1]$, throttle $[0, 1]$, and brake $[0, 1]$. The agent's action space is $\mathbf{a} \in [-1, 1]^2$, where the two components correspond to steering and acceleration. Negative acceleration implies braking, preventing simultaneous acceleration and braking (PETRAZZINI; ANTONELO, 2021).

While the network representing the agent outputs an energy value in IBC, which depends not only on an observation but also on an *action input*, in BC the output models a Beta distribution, which is naturally bounded and avoids manual clipping or squashing (PETRAZZINI; ANTONELO, 2021).

The policy network π_θ in BC outputs α and β , shaping the Beta distribution to adapt to diverse driving scenarios.

5.1.4 Data Augmented IBC (DA-IBC)

In Florence *et al.* (2022), for each observation x_i in a batch, the set of negative counter-examples $\{\tilde{y}_j^i\}_{j=1}^{N_{\text{neg}}}$ is generated by uniformly sampling from the action space: $\tilde{y}_j^i \sim U(y_{\min}, y_{\max})$. In this work, it is proposed modifying the sampling process for counterexamples for the loss in (17) of IBC. Instead of uniform sampling, it is generated negative counter-examples by sampling with replacement from the expert, the sampling is made from a random minibatch of actions from the expert dataset and subsequently adding Gaussian noise drawn from $\mathcal{N}(0, \sigma^2 I)$ to introduce perturbations. The developed method is called *Data-Augmented IBC (DA-IBC)*.

The set $\{\tilde{y}_j^i\}_{j=1}^{N_{\text{neg}}}$ is generated randomly and dynamically within the training loop for each observation x_i in a batch $\mathcal{B} = \{(x_i, y_i)\}$ from the expert dataset $\mathcal{D}$. As a result, different sets of counter-examples are created in each training iteration. Note that the actions in the set $\{\tilde{y}_j^i\}_{j=1}^{N_{\text{neg}}}$ are not dependent on any particular value of x_i . If the number of available expert actions is smaller than N_{neg} , we sample with replacement until the desired number is reached. Finally, each action in the set is perturbed with Gaussian noise to form the negative counter-examples.

This modification enhances the agent's learning process. Uniform sampling often generates unrealistic counterexamples with excessively high energy values, leading to a flat InfoNCE loss and stalled learning. In contrast, sampling from the expert dataset ensures that negative samples are more realistic, allowing the agent to learn finer distinctions between expert and counterexample actions.

Moreover, counterexamples sampled uniformly tend to be assigned near-zero probability, causing the model to focus disproportionately on positive samples, which stops improvement. However, by sampling from the expert dataset while introducing slight noise to negative samples alleviates this problem.

5.1.5 Adapted Derivative-Free Optimization (DFO)

Also the DFO algorithm is adapted from (FLORENCE *et al.*, 2022), presented in Algorithm 2, to perform implicit inference $\hat{y} = \arg \min_y E_\theta(x, y)$ with a trained EBM network. In DFO, the samples $\{\tilde{y}_i\}$ are initialized by sampling from a categorical distribution of the expert actions with probabilities equal to the corresponding weights $\{\tilde{w}_i\}$, with $w_i = \frac{1}{\hat{g}(y_i^{\text{expert}})}$ obtained by KDE. By using the inverse of the kernel density estimation of the actions as weights, we give more importance to less frequent actions. This approach can help the model learn better on underrepresented parts of the action space.

After the N_{iters} iterations of the method, instead of returning the action with maximum probability, a sample is returned by drawing from a categorical distribution of the actions $\{\tilde{y}_i\}$ and its respective probabilities $\{\tilde{p}_i\}$ found in the last iteration. This modification enhances the agent’s potential to perform multimodal behavior.

Algorithm 2: Adapted Derivative-Free Optimizer

Result: $\hat{y}$
Initialize: $\{\tilde{y}_i\}_{i=1}^{N_{\text{samples}}} \sim \text{Categorical}(\{y_i^{\text{expert}}\}, \{w_i\})$, $w_i \propto \text{KDE}(y_i^{\text{expert}})$, $\sigma = \sigma_{\text{init}}$
for $\text{iter} = 1, 2, \dots, N_{\text{iters}}$ **do**
 $\{E_i\}_{i=1}^{N_{\text{samples}}} = \{E_\theta(\mathbf{x}, \tilde{y}_i)\}_{i=1}^{N_{\text{samples}}}$ (Compute energies);
 $\{\tilde{p}_i\}_{i=1}^{N_{\text{samples}}} = \left\{ \frac{e^{-E_i}}{\sum_{j=1}^{N_{\text{samples}}} e^{-E_j}} \right\}$ (Softmax probab.);
if $\text{iter} < N_{\text{iters}}$ **then**
 $\{\tilde{y}_i\}_{i=1}^{N_{\text{samples}}} \sim \text{Multinomial}(N_{\text{samples}}, \{\tilde{p}_i\}_{i=1}^{N_{\text{samples}}}, \{\tilde{y}_i\}_{i=1}^{N_{\text{samples}}})$ (Resample with replacement);
 $\{\tilde{y}_i\}_{i=1}^{N_{\text{samples}}} \leftarrow \{\tilde{y}_i\}_{i=1}^{N_{\text{samples}}} + \mathcal{N}(0, \sigma)$ (Add noise);
 $\{\tilde{y}_i\}_{i=1}^{N_{\text{samples}}} = \text{clip}(\{\tilde{y}_i\}_{i=1}^{N_{\text{samples}}}, y_{\min}, y_{\max})$
 $\sigma \leftarrow K \sigma$ (Shrink sampling scale);
end if
end for
Return: $\hat{y} \sim \text{Categorical}(\{\tilde{y}_i\}, \{\tilde{p}_i\})$

5.2 EXPERIMENTS

This section presents a series of experiments designed to evaluate the proposed DA-IBC method in comparison with Behavioral Cloning (BC) and standard Implicit Behavioral Cloning (IBC).

The experimental setup was structured to address specific research questions regarding (1) route adherence and traffic light compliance under deterministic conditions, (2) multimodal decision-making in the absence of predefined navigation cues, (3) the emergence and consistency of multimodal behaviors across repeated trials, and (4) the robustness of learned policies in dynamic scenarios involving other vehicles and pedestrians.

To ensure comparability, in each experiment the agents were trained using the same expert demonstration dataset collected in the *Town01* environment for the predetermined conditions, while evaluations were carried out in *Town02*, providing information that tests the generalization capabilities of the models. Performance was assessed through several complementary metrics, including normalized distance traveled, compliance with traffic regulations, and collision rates with dynamic agents.

The following subsections describe the experimental setups and results in detail. The first experiment isolates traffic light compliance in a controlled scenario with a fixed route. The second introduces route ambiguity to assess multimodal decision-making, removing route information, and maintaining the traffic lights. The third maintains the conditions of the second experiment, but focuses on quantifying multimodality by measuring behavioral diversity under identical initial conditions. Finally, the fourth experiment evaluates performance in environments populated with dynamic obstacles such as vehicles and pedestrians, emphasizing safety and interaction-aware behavior.

To quantitatively compare the performance of the different methods, a composite metric that accounts for agent infractions is employed. The metric formulation is given in Equation 21:

$$\text{Score} = D \cdot 0.7^{N_{tl}} \cdot 0.5^{N_{vc}} \cdot 0.3^{N_{pc}}, \quad (21)$$

where D is the traveled distance, N_{tl} is the number of infractions in traffic lights, N_{vc} is the number of infractions due to vehicle collision, and N_{pc} is the number of infractions due to pedestrian collision. The exponential penalty factors are designed to reflect the relative severity of each type of infraction, with pedestrian collisions being penalized most heavily, followed by vehicle collisions, and traffic light violations being considered the least severe.

This metric was originally proposed in the NoCrash benchmark, as presented by (ZHANG, Zhejun *et al.*, 2021), and later adopted by the CARLA Leaderboard. The success rate is defined as the percentage of routes completed without collisions or blockages. However, in our experiments the predefined route constraint was removed; therefore, the driving score is defined as the product of the traveled distance and an infraction penalty, which acts as a discount factor aggregating all triggered infractions. For example, if an agent runs two red lights during a route and the penalty coefficient for running a single red light is 0.7, the resulting infraction penalty would be 0.7^2 . The driving score provides a fine-grained evaluation by accounting for multiple types of infractions and is better suited for assessing long-distance driving behavior.

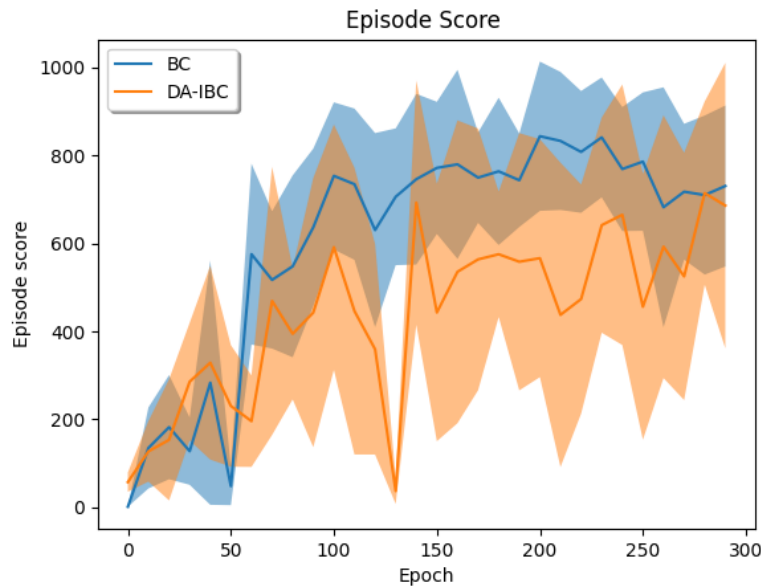
5.2.1 Traffic lights with predetermined route

In the first experiment, the BEV input is composed only by the road, route and lane channels. The behavior of the BC and DA-IBC agents was evaluated under con-

trolled conditions. Each agent was required to follow a predefined route; deviation from this route resulted in the immediate termination of the episode. Agents were also required to obey traffic signals and to remain within the lane boundaries. To isolate the effects of route adherence and traffic signal compliance, pedestrians and other vehicles were disabled during this experiment.

Two agents were trained for each method (BC and DA-IBC) over 300 epochs using data collected in the *Town01* environment, with output actions modeled using a Beta distribution. The evaluation was conducted in *Town02* environments of the CARLA simulator, as shown in 10. The episode score was defined as the product of the distance traveled and a penalty factor related to traffic light compliance, due to the absence of pedestrians and vehicle the N_{pc} and N_{vc} were set to 0. For each method, the average and standard deviation are computed over two different agents evaluated ten times for 3,000 steps (or 300s). This evaluation is done each 10 training epochs.

Figure 10 – Training curves of BC and DA-IBC over 300 epochs, trained on Town01 and evaluated on Town02, reporting mean and standard deviation of the distance-based score with traffic light penalty, computed from multiple evaluations every 10 epochs.



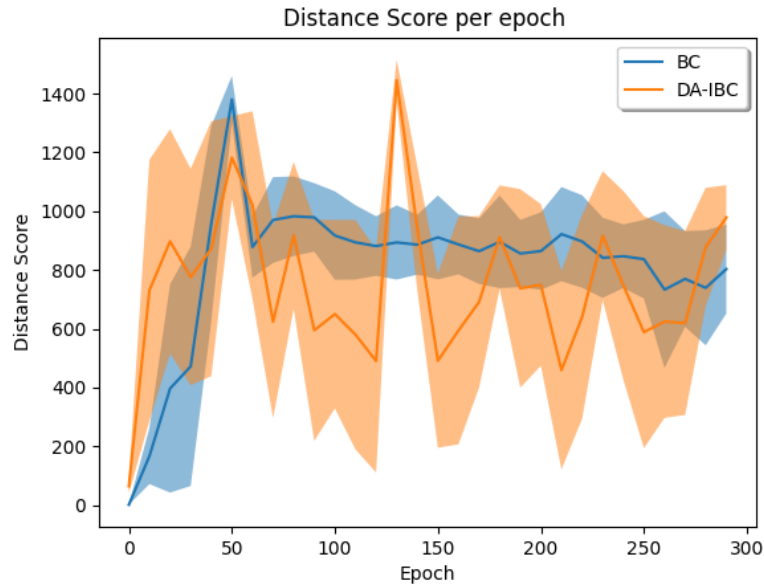
During the initial stages of training, the DA-IBC method achieved faster learning progress compared to BC, which required a greater number of epochs to converge toward stable performance. However, the overall scores achieved by DA-IBC were consistently lower than those of BC. Video analysis further indicated that DA-IBC generally complied with traffic lights and maintained adherence to lane and route boundaries. However, in certain intersections, the agent exhibited difficulty resuming movement when the signal transitioned to green. The BC agent demonstrated comparable compliance with traffic lights as well as road and route boundaries.

The distance scores are shown in Figure 11, the distance is calculated considering only the distance the agent traveled within the route. For each method, the average and standard deviation are computed over two different agents evaluated ten times for 3,000 steps (or 300s). This evaluation is done each 10 training epochs. Importantly, the premature stopping behavior of DA-IBC contributed to its reduced distance scores.

The result of the traffic light compliance is shown in Figure 12. The results are the ratio $\frac{R_{tl}}{E_{tl}}$ (where E_{tl} is the number of encountered traffic lights and R_{tl} is the number of respected traffic lights). Overall, both agents exhibited similar levels of adherence to traffic signals.

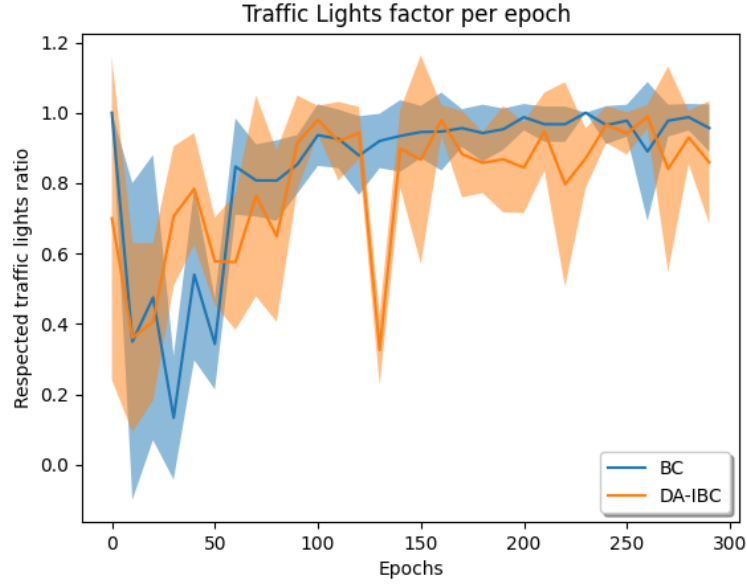
The DA-IBC agent failed to respect traffic lights primarily in scenarios where the signal turned red while the agent was already approaching the intersection. In such cases, the agent was often unable to brake in time, resulting in stopping beneath the traffic light or within the intersection. The BC agent encountered fewer of these situations but displayed similar behavior when they occurred. In some instances, the BC agent completely ignored the red light, proceeding through the intersection despite the signal being closed.

Figure 11 – Training curves of BC and DA-IBC over 300 epochs, trained on Town01 and evaluated on Town02, reporting mean and standard deviation of the distance traveled of the agents, computed from multiple evaluations every 10 epochs.



To better understand the behavior of the agent in intersection with predetermined routes, the energy map for this situation is analyzed. In fig. 13 it is possible to observe that the agent in the intersection, even with the option of going straight, turned right due to the predetermined route. The lower area of the heatmap has lower energy values, resulting in turning left, indicating that it did not consider the possibility of going out of

Figure 12 – Training progress of BC and DA-IBC over 300 epochs, trained on Town01 and evaluated on Town02, reporting mean and standard deviation of the traffic light compliance ratio $\frac{R_{tl}}{E_{tl}}$ (where E_{tl} , computed from multiple evaluations every 10 epochs).



the route.

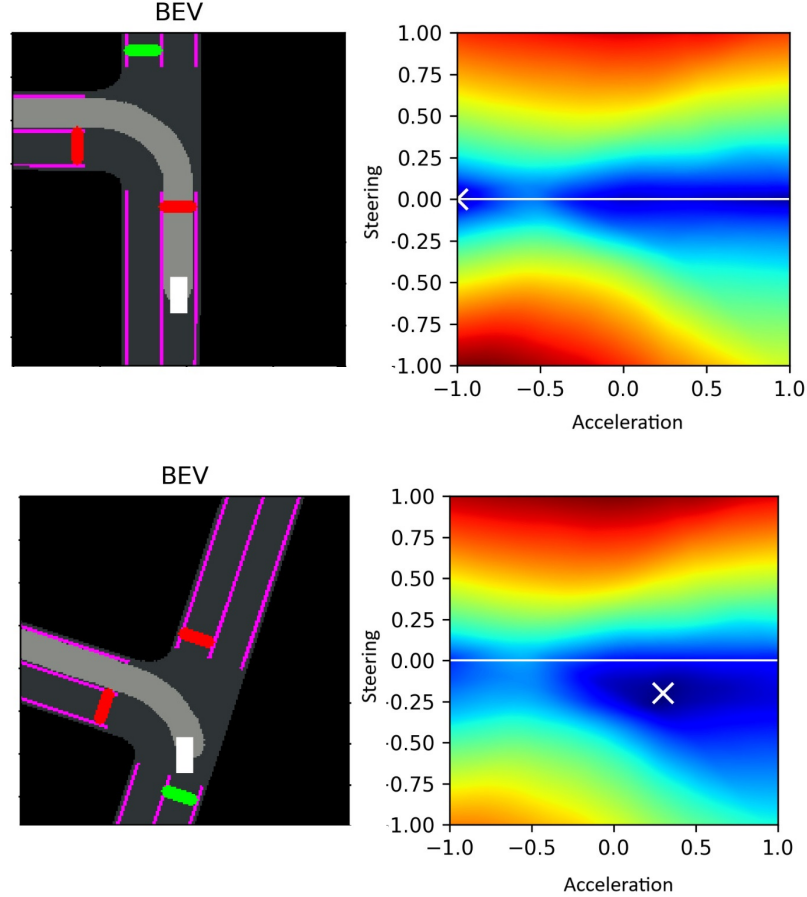
5.2.2 Traffic lights without predetermined route

This experiment was designed to evaluate and compare the performance of BC, IBC, and DA-IBC in executing fundamental driving tasks—specifically, maintaining lane and road adherence and complying with traffic light signals. The primary objective was to assess how each method handles multimodal decision-making. To this end, the route channel was removed from the BEV input, compelling the agent to independently determine its direction at intersections without external guidance.

To this end, three agents were trained using data collected in the *town01* environment of the CARLA simulator (Figure 8), with the proposed DA-IBC, IBC and BC with the outputs modeling a Beta distribution. The dataset comprised demonstrations over 10 distinct driving routes, of which 8 were used for training and 2 were held out for validation. All agents were trained offline on the same data to ensure a fair comparison.

In the evaluation each agent was deployed in the *town02* environment for a fixed duration of 300 seconds. Importantly, no route information was provided during inference. This constraint required the agents to make real-time decisions at intersections without prior knowledge of the correct path, thereby simulating a forced multimodal scenario in which multiple plausible expert-like behaviors could emerge. Such a setup challenges the policy to not only react to the current state but also implicitly infer a reasonable driving intent under ambiguity.

Figure 13 – Left: BEV scene perceived by the agent. Right: energy landscape $E(s, a)$ over steering and acceleration, with the zero-steering line indicated in white. Top: agent stopped at a red traffic light; bottom: agent accelerating and turning left after the light turns green. Cross markers denote actions inferred by ten DFO runs for the same BEV input.

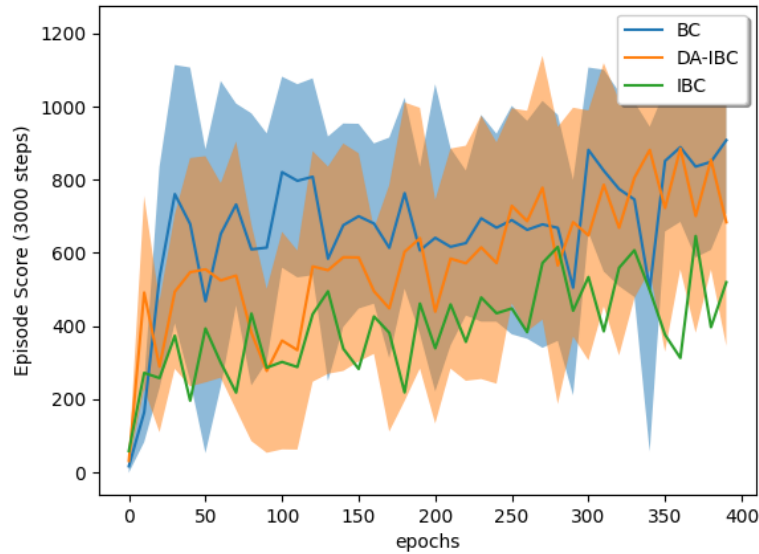


It was found that distance traveled alone was not a reliable performance metric, as it was heavily influenced by the number and timing of traffic light encounters during the episode. Agents that stopped at red lights, as they should, would naturally travel shorter distances compared to those that encountered fewer signals or failed to stop appropriately. Thus, variations in total distance did not necessarily reflect behavioral quality or success in policy learning.

In Fig. 14, this score is shown for all three methods, where the average was computed for two agents per method and over 10 episodes each of 3,000 timesteps for each agent. The episode score was defined as the product of the distance traveled and a penalty factor related to traffic light compliance, due to the absence of pedestrians and vehicle the N_{pc} and N_{vc} were set to 0. Even though BC attains higher score faster than both IBC and DA-IBC, its learning problem is easier when compared to them, which need to learn an energy landscape as a function of the action instead of the average action taken at a particular observation as done in BC. Thus, it takes more time

to attain higher scores for both IBC and DA-IBC agents. In addition, IBC and DA-IBC learns multimodal action distributions, allowing multimodal behavior when generating vehicle trajectories, which is impossible for BC.

Figure 14 – Training curves of BC, IBC, and DA-IBC on Town01 with evaluation on Town02, showing the mean and standard deviation of the distance-based score with traffic-light penalty, evaluated every 10 epochs during offline training.



5.2.3 Multimodality test

The ability to produce diverse yet plausible driving behaviors under identical initial conditions, of the three agents (BC, IBC, and DA-IBC), varies significantly due to their underlying modeling assumptions. This experiment aimed to quantify that variability in behavior and explore how different policy formulations capture the ambiguity present in urban driving.

To conduct this evaluation, we defined a set of 10 distinct starting points within the *town01* environment. These locations were selected to include a variety of local road structures, such as intersections, curves, and straight segments, thereby ensuring a representative set of decision-making contexts. For each starting point, the agent was evaluated over a 30-second horizon, during which it was required to operate without any route input, forcing it to rely solely on its learned policy to select among multiple possible driving paths. Each of the 10 starting conditions was repeated five times, resulting in 50 total episodes per method.

The primary objective of this evaluation was to observe the extent to which different policies produce distinct and valid trajectories from the same starting position, which is a hallmark of multimodal behavior. All agents were deployed in the same

simulation conditions to ensure fairness. The variability of generated trajectories was then analyzed both qualitatively and quantitatively.

The results of this experiment are summarized in Table 5.2.3, which reports trajectory divergence metrics across rollouts for each method.

Table 2 – Behavior comparison of trained BC, IBC, and DA-IBC agents in Town02, reporting the frequency of path choices at two-option intersections over multiple episodes from 10 different starting points.

	BC	IBC	DA-IBC	Intersection Type
Intersection	↑ ← →	↑ ← →	↑ ← →	
Start Point 1	5 0 0	1 4 0	3 2 0	Left/Straight
Start Point 2	5 0 0	0 5 0	4 1 0	Left/Straight
Start Point 3	5 0 0	2 2 0	5 0 0	Left/Straight
Start Point 4	2 0 3	0 0 5	0 0 5	Right/Straight
Start Point 5	0 0 5	0 0 5	1 0 4	Right/Straight
Start Point 6	0 0 5	0 0 5	0 3 2	Left/Right
Start Point 7	0 0 5	0 0 5	2 3 0	Right/Straight
Start Point 8	0 0 5	0 1 4	0 4 1	Left/Right
Start Point 9	0 0 5	0 0 5	2 0 3	Right/Straight
Start Point 10	0 0 5	0 0 5	4 1 0	Left/Straight

As expected, Behavioral Cloning (BC) exhibited minimal variability, typically repeating the same trajectory across all episodes from a given starting point. At intersections where a right turn was available (episodes 4 to 10), the BC agent chose to turn right in 94% of the cases. When no right turn option was present (episodes 1 to 3), the agent proceeded straight in 100% of the trials and never selected a left turn.

In contrast, Implicit Behavioral Cloning (IBC) showed notable limitations. In episodes 4 to 10, the agent became biased toward right turns, selecting this action in 97% of the cases. However, in the starting points 1 to 3, the agent exhibited some degree of multimodality, proceeding straight in 73% of the trials and turning left in 20%, while the remaining 7% resulted in off-road behavior.

DA-IBC produced a substantially broader range of behavior patterns while maintaining adherence to traffic rules and road geometry, highlighting its ability to represent multimodal decision-making in ambiguous scenarios. In the starting points 4 to 10, DA-IBC distributed its decisions across turning right (42%), turning left (31%), and going straight (24%).

Interestingly, an exception was observed at starting point 4, where BC occasionally deviated from its otherwise rigid, the video showed that the agent started to move right and redirected to go straight. In three out of five evaluation runs, the agent chose to turn right instead of going straight. However, in those cases, it failed to stop at a red traffic light, suggesting that while the deterministic policy may occasionally escape its

typical mode due to subtle variations in input or internal state, such deviations often lack the robustness or intent structure required to maintain safe driving behavior.

In contrast, IBC agent with uniform sampling demonstrated greater behavioral variance. This agent showed a noticeable bias toward turning right when that option was available, likely a consequence of the high energy counterexamples leading to less refined actions and high probability concentrated on the same action, also explored left and off-road paths in scenarios where a straight path was not clearly favored. While the increased diversity in action selection confirms IBC's ability to represent multiple plausible behaviors, the lack of structure in the sampling occasionally resulted in suboptimal or unsafe choices.

The DA-IBC agent, which samples are based on a learned distribution of expert examples, displayed the most coherent and realistic multimodal behavior. It frequently produced different trajectories from the same initial conditions, indicating that it had learned to capture a better representation of the underlying distribution of expert behaviors. This is likely facilitated by the energy-based model (EBM) structure, which provides a smooth energy landscape over possible actions. As the model converges during training, distinct local minima emerge in the energy space, enabling the policy to sample from diverse but valid modes of behavior. This results in trajectories that not only vary meaningfully but also remain safe and rule-compliant.

Behavioral tendencies of the DA-IBC agent were consistent. In intersections offering a choice between left and straight, the agent most frequently selected the straight path, though left turns were occasionally chosen. In scenarios with straight or right options, it consistently favored going straight. Interestingly, in intersections with only left or right as options, the agent showed a preference for turning left, a pattern that can be attributed to the energy values produced by the model, which tended to follow the order: straight < left < right. This suggests that the learned energy landscape reflects a natural preference hierarchy, possibly inherited from biases in the expert demonstrations.

Also, in order to evaluate multimodal decision making, both BC and DA-IBC agents were set to navigate freely in *town02*, after their training with data from *town01*.

In Fig.15, the BC agent primarily follows a single trajectory through the town, with occasional deviations caused by stochasticity in the simulator. These deviations do not reflect true multimodal behavior, but rather arise from noise during action execution. The agent could not explore the whole city.

In contrast, as shown in Fig.16, the DA-IBC agent learns a genuinely multimodal policy that allows it to take distinct actions at the same decision points, enabling it to explore and cover the entire city. In both simulations, there were no crashes, but some traffic light infractions occurred.

We also investigated the energy landscape of the trained agent in three situations

Figure 15 – Trajectory of the BC agent in a 3,000 s simulation in Town02, showing limited exploration and lack of multimodal behavior without route guidance, resulting in confinement to a small city region.

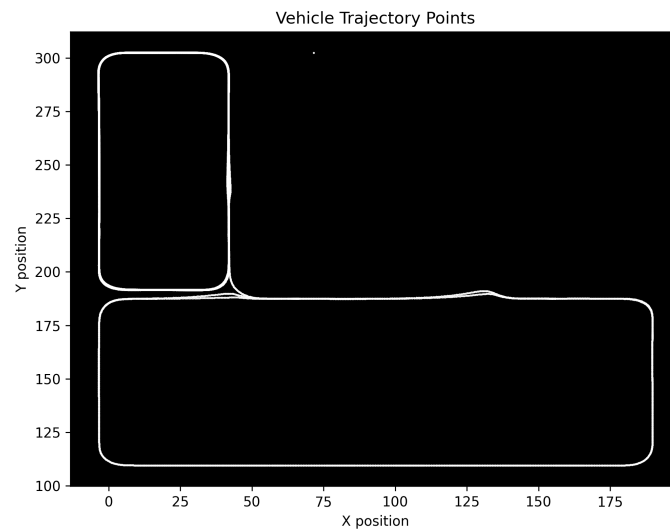


Figure 16 – Trajectory of the DA-IBC agent during a 3,000 s simulation in Town02, demonstrating learned multimodal behavior that enables diverse decisions at intersections and full city exploration without route guidance.

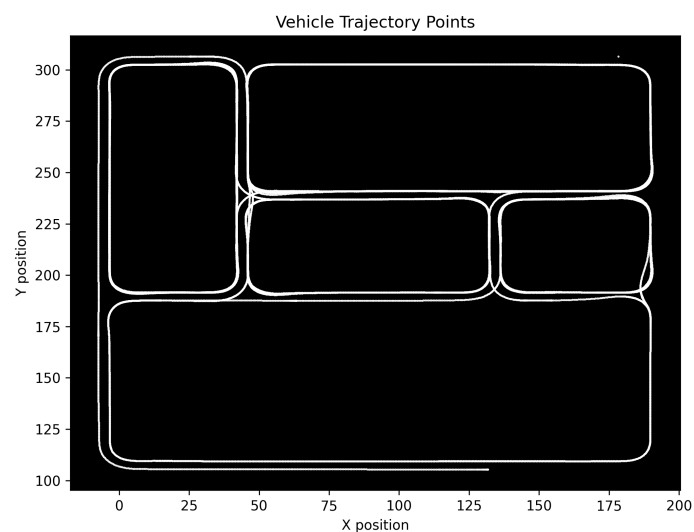


Figure 17 – Left: BEV scene perceived by the agent. Right: energy landscape $E(s, a)$ over steering and acceleration in a multimodal intersection scenario, with zero steering indicated. Top and bottom illustrate left and right turn solutions at the same intersection, with white points showing actions inferred from multiple DFO runs.

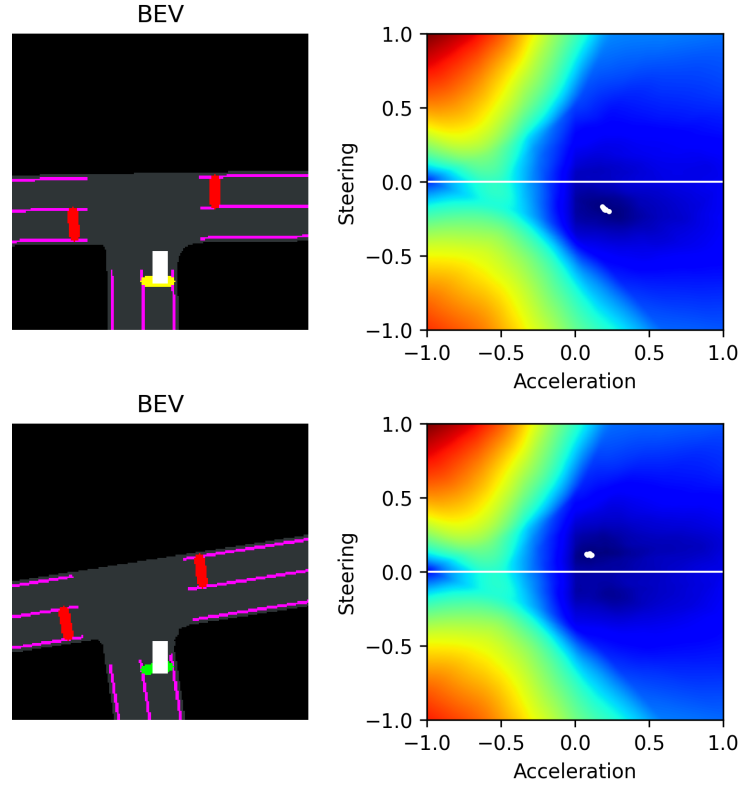
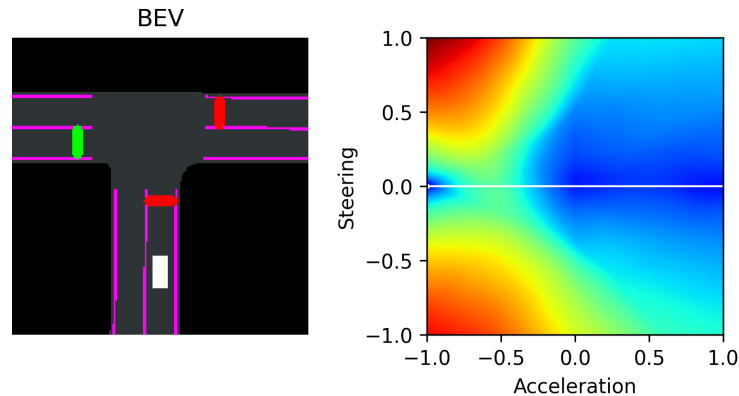


Figure 18 – Energy $E(s, a)$ as a function of the two-dimensional action a (right plot), in situations where the steering action distribution is nearly unimodal: stopping at a red light. The ten DFO inferences are at (0,0) for acceleration/steering with the fixed BEV as input (left plot).



in *town02*. In Fig. 17, we can see the agent deciding to turn left or right at the same T-intersection, and the corresponding energy function, where the blue color corresponds to lower energy. Ten inferences were made with the proposed DFO algorithm and plot as 10 white points in the energy landscape, for each case. The left image is the BEV scene as perceived by the agent.

In Fig. 18, the energy landscape for the vehicle stopping at the red light indicates a slightly different learned function, with the minimum at (0,0), which signifies zero acceleration and zero steering. These energy landscapes can work as probabilities distributions, with the actions that find minimums of the energy function being the most likely actions given the expert dataset, which contain possibly conflicting navigation strategies. Thus, as seen in Fig. 17, both turning behaviors were learned and modeled by the EBM even though no explicit command of turning left or right were present in the expert training set.

These behaviors are showed in two videos showcasing the same DA-IBC agent selecting distinct routes through the same intersection, demonstrating its capacity for real multimodal decision-making, are available at the following links: [Video 1](#) and [Video 2](#).

5.2.4 Vehicles and pedestrians

In the following experiment, the absence of a predetermined route was maintained and dynamic obstacles introduced, specifically other vehicles and pedestrians, that the agent must perceive and respond to appropriately. To enable this, the input representation is augmented with six additional channels, providing temporal information about these new elements in the environment.

Similar to previous experiments, two agents were trained for 600 epochs using data collected in the *town01* environment of the CARLA simulator, with the proposed DA-IBC and BC with the outputs modeling a Beta distribution. The split of training and testing routes were 8 and 2 respectively. The evaluation was fixed in 300s duration in an endless environment and the evaluation happened every 40 epochs.

For vehicle representation, three channels are added, each corresponding to the positions of surrounding vehicles at different past timesteps: 16, 9, and 1 steps prior to the current observation. Similarly, three additional channels are used to represent the positions of pedestrians at the same three previous timesteps.

This temporal encoding is crucial, as it allows the agent to infer the motion patterns and trajectories of both vehicles and pedestrians. Such information enables the model to anticipate potential collisions and to learn behaviors that ensure safe and socially compliant navigation, such as slowing down near crosswalks or avoiding unsafe proximity to moving vehicles. By incorporating these new input modalities, the learning framework is better equipped to handle complex, real-world scenarios that involve dynamic interactions with other road users.

To assess the impact of incorporating dynamic obstacles into the learning process, we reproduce the initial experiment described in the previous section. Specifically, we evaluate the agents across two of the same set of metrics: normalized distance traveled and traffic light respect ratio. These metrics allow us to compare the overall

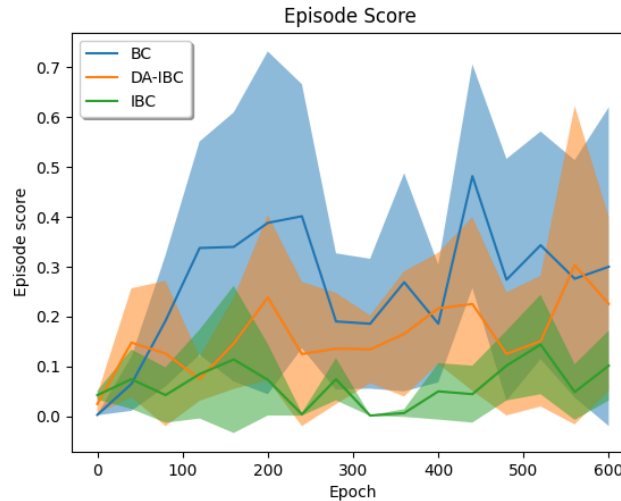
behavioral performance of the agents across training epochs under more complex and realistic conditions.

The episode score is now expanded considering the pedestrians and vehicles-related infractions. Each infraction penalizes the agent reducing the score. Additionally, we introduce new evaluation metrics focused on safety: the collision rate with vehicles and pedestrians. This metric quantifies the agent’s ability to avoid dynamic obstacles in its environment, providing a direct measure of its learned capacity for socially compliant and collision-free navigation.

By analyzing these four performance indicators over the course of training, we aim to observe whether the agent progressively acquires safer and more reliable behaviors in the presence of other road users. The results are visualized in the corresponding plots, which illustrate how performance evolves as the model is exposed to increasingly sophisticated scenarios.

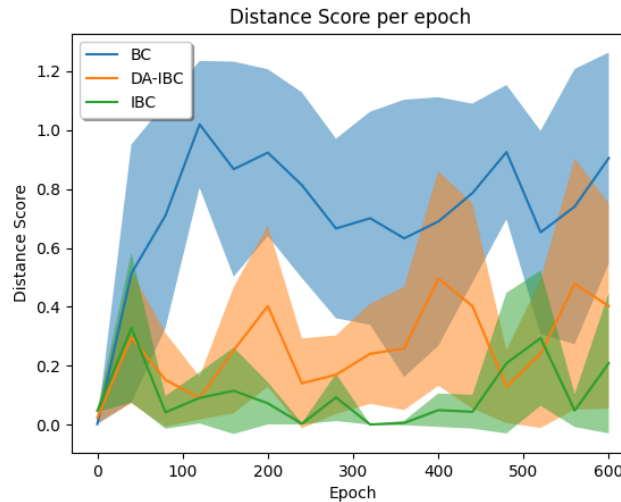
In Figure 19, the overall score is presented for BC, IBC and DA-IBC agents. The score is the normalized distance traveled multiplied by the traffic light, pedestrians and vehicles infractions, $(0.7^{ntl} * 0.2^{npc} * 0.5^{nvc})$, where ntl is the number of traffic light infractions, npc is the number of collisions with pedestrians and nvc is the number of collision with vehicles in the episode of 300s). For each method, the average and standard deviation are computed over two different agents evaluated ten times for 3,000 steps (or 300s). This evaluation is done each 10 training epochs. Throughout most of the training process, BC achieved higher scores, except at epoch 550, where DA-IBC outperformed it, while vanilla IBC achieved low score results.

Figure 19 – Training curves of BC, IBC and DA-IBC on Town01 with evaluation on Town02, reporting mean and standard deviation of the normalized distance-based score weighted by traffic-light and collision penalties, evaluated every 10 epochs.



To better understand this result, it is important to analyze each evaluation metric

Figure 20 – Normalized comparison of traveled distances for BC, IBC and DA-IBC agents in the same training and evaluation conditions previously determined.



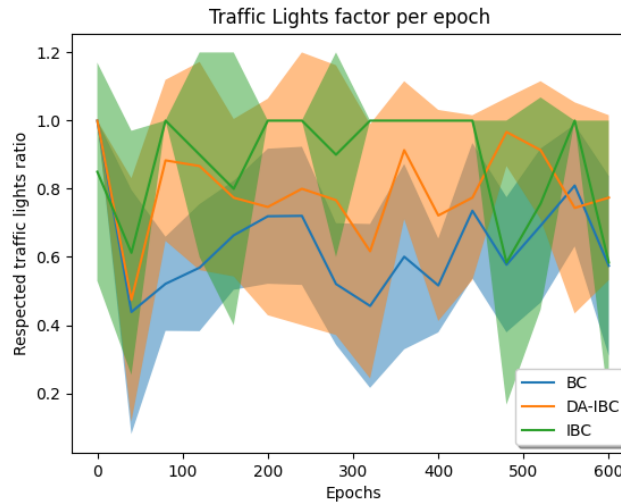
individually, as the overall scores are strongly influenced by the number of infractions and the distance traveled. As shown in Figure 20, the BC agent consistently achieved greater distances compared to IBC and DA-IBC agents. However, DA-IBC presented a better overall result when compared to IBC. Analyzing the videos from IBC agent, it presented a recurring behavior, it stopped near traffic lights and was not able to continue moving.

Upon analyzing both the quantitative results and the qualitative behaviors observed in the recorded simulation videos, several notable patterns emerged. A significant reduction in traveled distance was observed in all agents compared to the previous experiment. This decline is attributed to the agents now stopping in response to the presence of vehicles and pedestrians, indicating an improved ability to respect dynamic obstacles and avoid collisions.

Despite this improvement, neither agent demonstrated perfect compliance with traffic lights, as infractions were still recorded. However, DA-IBC exhibited better overall adherence than IBC and BC. Video analysis at the 400 epoch training mark revealed distinct behavioral differences between the agents. The BC agent achieved longer travel distances primarily because it failed to stop even after committing infractions, frequently ignoring red traffic lights. In contrast, the DA-IBC agent showed more consistent compliance with traffic signals, as illustrated in Figure 21.

Regarding infractions involving pedestrians and vehicles, the BC and IBC agents, despite recording relatively few collisions, often continued to drive after an accident. In some episodes, it even went out off the road boundaries following a collision. In contrast, the DA-IBC agent consistently stopped moving after colliding with a pedestrian or vehicle and remained stationary for the remainder of the episode. Additionally, in

Figure 21 – Comparison of traffic light compliance ratios for BC, IBC and DA-IBC agents, for 600 epochs of training progress, the traffic light factor is the number of respected traffic lights divided by the total of traffic lights encountered.



situations where the agent had already crossed a traffic light and a pedestrian or vehicle entered the intersection simultaneously, DA-IBC successfully avoided a collision but chose to stop and did not resume driving. This behavior suggests a conservative strategy, where DA-IBC preferred halting over exposing itself to further risk. As a result, DA-IBC achieved lower infraction scores related to pedestrians and vehicles, as illustrated in Figures 22 and 23.

Qualitative analysis of the evaluation videos also revealed that when agents encountered both a traffic light and nearby vehicles simultaneously, they tended to prioritize vehicle avoidance while neglecting traffic signals often proceeding through red lights without stopping. Conversely, in scenarios without surrounding vehicles or pedestrians, the agents generally adhered to traffic signals correctly. Moreover, on straight roads with multiple obstacles, the agents demonstrated improved behavior, braking appropriately to avoid pedestrians and vehicles when necessary.

Quantitative results are reported for the 10 episodes corresponding to the epoch with the highest episode score for each agent. These results are presented in Tables 3, 5, and 4.

It can be observed that both BC, IBC and DA-IBC learned to respect other vehicles in most cases, colliding with a vehicle in few episodes each. Furthermore, video analyses revealed that, despite multiple encounters with pedestrians, all the agents were generally able to navigate without incidents. However, for BC the presence of other vehicles at intersections with traffic lights led to a higher number of traffic light infractions. In some particular cases, the agent passed through a red light because the signal changed while the agent was already crossing the intersection. In the remaining

Figure 22 – Vehicle Collision Score considering the number of collision with vehicles divided by the distance traveled, for BC, IBC and DA-IBC for 600 epochs of training progress.

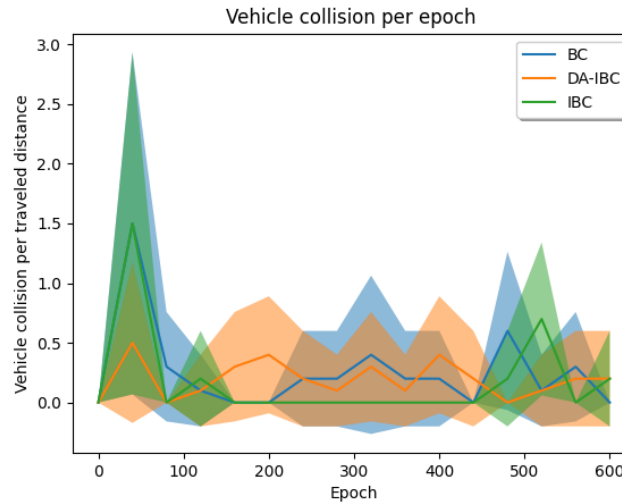


Figure 23 – Pedestrian Collision Score considering the number of collision with vehicles divided by the distance traveled, for BC, IBC and DA-IBC for 600 epochs of training progress.

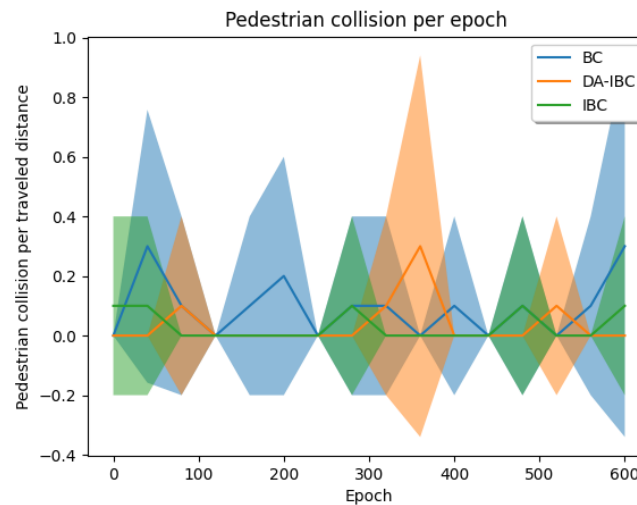


Table 3 – BC evaluation results over 400 training epochs, reporting per-episode collisions with vehicles and pedestrians, red-light infractions, and distance traveled across 10 episodes.

Episodes	Vehicle Infraction	Pedestrian Infraction	Red lights violated	Distance [m]
1	0	0	0	747
2	0	0	2	487
3	0	0	2	504
4	0	0	2	792
5	0	0	3	505
6	0	0	0	175
7	0	0	1	320
8	0	0	1	808
9	1	0	1	668
10	0	0	2	747

Table 4 – IBC evaluation results over 520 training epochs, reporting per-episode collisions with vehicles and pedestrians, red-light infractions, and distance traveled across 10 episodes.

Episodes	Vehicle Infraction	Pedestrian Infraction	Red lights violated	Distance [m]
1	0	0	0	52
2	1	0	0	175
3	0	0	0	26
4	1	0	1	311
5	1	0	0	362
6	2	0	2	367
7	0	0	1	78
8	0	0	1	274
9	1	0	0	66
10	1	0	1	637

cases, the agents failed to recognize and appropriately react to the traffic signals.

Regarding the traveled distance, several considerations must be made. The distance was significantly impacted by traffic jams near intersections, where agents often had to wait in queues. Video analyses showed that in some episodes BC agent drove off the road in intersections, where it took the mean path and ended out of the road. Also, in intersections with a high density of vehicles, the BC agent became confused and left the roadway. IBC while taking decision proceeded to drive off the road in intersection and also stopped suddenly, similar to DA-IBC but more frequently. In

Table 5 – DA-IBC evaluation results over 4560 training epochs, reporting per-episode collisions with vehicles and pedestrians, red-light infractions, and distance traveled across 10 episodes.

Episodes	Vehicle Infraction	Pedestrian Infraction	Red lights violated	Distance [m]
1	0	0	0	931
2	0	0	0	13
3	1	0	0	178
4	1	0	1	778
5	1	0	0	106
6	0	0	3	890
7	0	0	1	141
8	0	0	0	65
9	0	0	1	233
10	0	0	1	482

Table 6 – Mean value of tables 5, 4 and 3 and result of the score for the calculated means

Method	Vehicle Infraction	Pedestrian Infraction	Red lights violated	Distance [m]	General Score
DA-IBC	0,3	0	0,7	381,7	211,12
IBC	0,7	0	0,6	234,8	120,68
BC	0,1	0	1,4	575,3	210,36

contrast, the DA-IBC agent consistently remained within the road boundaries. In some instances the agents suddenly stopped moving near intersections. BC exhibited fewer instances of stopping, while IBC and DA-IBC stopped more often. Further analysis indicated that most of these stops occurred when, while crossing an intersection, the agent needed to brake due to other vehicles or a closing traffic light. In such situations, the agent was unable to determine the next course of action and consequently came to a halt.

To better visualize the differences in each method, we take the mean of the results of each agent in the 10 episodes, these results are shown in table 6. Also, in the last column the general score is calculated the same of Fig. 19.

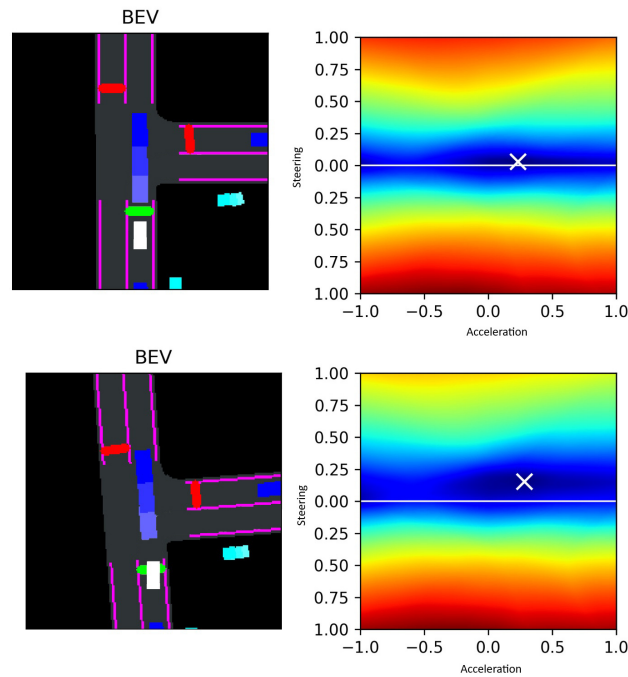
In terms of vehicle infractions, DA-IBC achieved a 42% reduction compared to IBC. Regarding red light violations, DA-IBC reduced the number of infractions by 50% relative to BC. Additionally, DA-IBC traveled a distance 62% greater than that achieved by IBC. Overall, DA-IBC attained a higher general performance score in 75% of the episodes compared to IBC and in 1% of the episodes compared to BC.

The improvement of DA-IBC over IBC stems from modifications to the infer-

ence process, in which counterexamples are dynamically generated and selected using KDE-based weighting to favor more realistic actions. These counterexamples directly influence the contrastive energy-based loss, as harder and more plausible negatives contribute more strongly to the optimization. Consequently, the loss enforces a sharper separation between expert actions and counterexamples in the energy landscape, enhancing the agent's ability to precisely discriminate expert behavior while reducing trajectory overfitting and preventing collapse to a single path.

The result indicate that even if the distance traveled by the BC agent is higher, it committed more infractions, which reduces significantly the general score, the final score is nearly the same as DA-IBC, indicating that the method has promising results. However, IBC achieved a lower score, this indicates that the alteration in the inference improved the method.

Figure 24 – Top left: BEV image perceived by the agent, while other dynamic obstacles are near (pedestrians and vehicles). Top right: energy landscape $E(s, a)$ of the agent interacting with the environment, where the gradient goes from red (high energy - less likely) to blue (low energy - most likely). Bottom-left: BEV image of the same agent 10 steps further, passing the traffic light. Bottom-right: Energy function of the agent deciding to turn right.



We also investigated the energy landscape of the trained agent in three situations in *town02*. In Fig. 24, while the car is before the traffic light, the agent lowest energy on zero for steering and dislocated to the right indicating that the agent understood that the traffic light is green and it needed to start moving. Past the traffic light the energy map changes, the lowest point now is in a positive value for steering indicating that the car is turning right and the acceleration continues the same.

Figure 25 – Top left: BEV image of the agent in the road, while a pedestrian is in the road. Top right: Energy function for the agent in a safe distance from the pedestrian still accelerating. Bottom-left: BEV image of the same agent 2 steps further, closer to the pedestrian. Bottom-right: Energy function of the agent braking to avoid colliding, the "x" mark, is in the point of zero steering and negative acceleration.

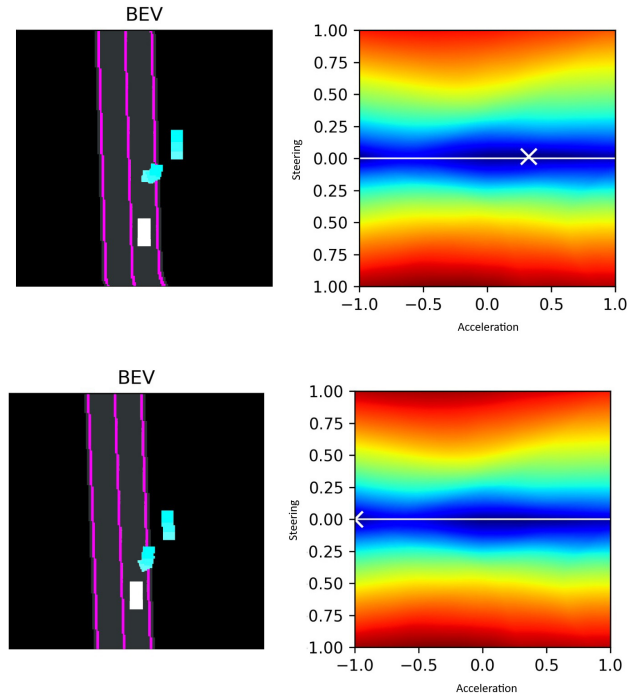
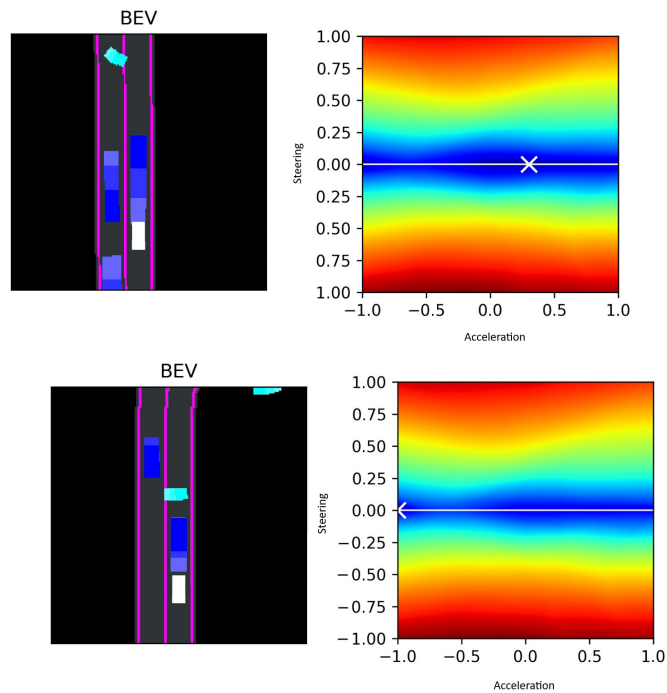


Figure 26 – Similar situation and behavior from 25, but now the dynamic agent is a vehicle.



In Fig. 25 and Fig. 26 the scenarios are similar, there is a dynamic obstacle on the road blocking the way. In both scenarios the agent is accelerating constantly without steering, however, when the agent approaches the pedestrian or the stopped car, the agent instantly go to negative acceleration, indicating that the agent hit the brakes.

It is important to notice that the map contains more than one blue spot indicating different minimum points, however the lower the energy the higher the probability for the action to be chosen. This behavior was learned purely by the interaction with the energy function, which means that there were no external inputs to indicate the direction that the agent should follow. The agent was able to maintain its multimodality while interacting with other obstacles.

6 CHALLENGES AND FUTURE WORKS

While evaluating the results for DA-IBC in every experiment and analyzing the videos to check its behavior, some patterns were noticed, that needs to be further improved.

In all approaches that had traffic lights occasional infractions were observed in both approaches. These incidents typically occurred when a traffic signal transitioned to red while the vehicle was already in close proximity to the intersection, leaving insufficient time or distance for the agent to safely stop. Such cases suggest that the agents may have limited capacity for anticipatory behavior under tight timing constraints, likely due to the lack of explicit future intent modeling or temporal planning in the policy architecture.

Also, in more complex scenarios involving multiple dynamic entities (vehicles, pedestrians, and traffic lights), the DA-IBC agent occasionally came to a stop in the middle of the intersection, indicating confusion in handling multi-constraint environments.

For instance, the BC agent, even after 280 training epochs, frequently ignored traffic light signals, proceeding through intersections without stopping as required. On the other hand, the DA-IBC agent, while generally more cautious around vehicles, often failed to account for pedestrians, especially when they were present at crosswalks or crossing intersections.

Both agents also exhibited a recurrent suboptimal behavior when encountering a vehicle and a traffic light simultaneously. In these situations, they tended to overcompensate by stopping for prolonged periods, often longer than necessary. After stopping, they resumed acceleration only after a noticeable delay in the simulation timeline, reflecting a lack of refined decision-making in scenarios involving multiple simultaneous constraints.

These findings highlight the limitations of both learning approaches under increased environmental complexity, especially when the agent must process and respond to several interacting elements simultaneously. This underscores the need for further improvements in perception integration and decision-making under uncertainty.

There were epochs when the agent respect both without causing accidents. However, some cases highlighted further limitations in the learned policies: when a traffic light turned green, but a vehicle was still crossing the intersection, the agent frequently stopped abruptly in the middle of the road and failed to continue moving, remaining stationary for the remainder of the episode or for an extended duration. This behavior suggests a confused interpretation of priority and timing, where the agent is unable to reconcile conflicting signals, such as a green light coinciding with a moving obstacle, resulting in indecisiveness or overly cautious actions.

Beyond the qualitative inspection of agent behavior, future work should place

stronger emphasis on a more comprehensive validation and quantitative analysis of the proposed DA-IBC framework. While the current evaluation protocol captures overall performance trends and safety-related outcomes, a finer-grained analysis is required to better understand the conditions under which the method succeeds or fails. This includes decomposing performance metrics by scenario type (intersections, pedestrian crossings, and high-density traffic), analyzing failure cases with respect to temporal proximity to events such as traffic light transitions, and measuring behavioral consistency across repeated trials. Additionally, statistical significance testing and ablation studies on key components such as data augmentation strategies, energy regularization, and inference procedures would provide deeper insight into the contribution of each design choice. Such an expanded quantitative evaluation would strengthen the empirical validation of DA-IBC and help guide future architectural and training improvements.

7 CONCLUSION

This work investigated the use of Implicit Behavioral Cloning (IBC) with Energy-Based Models (EBMs) to address the multimodality challenges inherent in autonomous driving. To this end, we proposed DA-IBC (Data-Augmented IBC), an extension of standard IBC that introduces two main innovations: modifying the inference DFO algorithm to initialize action samples from the expert dataset rather than from a uniform distribution, and enhancing the training loss by generating counterexamples through perturbations of expert actions, thereby refining the energy function to better distinguish expert behavior from nearby alternatives. While traditional Behavior Cloning (BC) often collapses to averaged actions in multimodal scenarios, EBMs provide a more expressive framework by learning an energy function over the action space, enabling diverse and adaptable decision-making.

In section 1.2 we present the research questions, that are answered below:

1. The results indicate that the IBC method is effective in controlling the agent in urban environments. However, further improvements through the DA-IBC framework are necessary to enhance the agent’s learning capabilities.
2. In terms of quantitative performance, DA-IBC achieved results comparable to those of BC when considering travel distance and traffic infractions, indicating that the introduction of multimodality does not compromise safety or stability. At the same time, DA-IBC provided greater behavioral diversity, an essential requirement for realistic autonomous driving in dynamic, uncertain environments.
3. As task complexity increased, all three agents BC, IBC, and DA-IBC exhibited limitations that warrant further investigation. Nonetheless, DA-IBC consistently outperformed BC in these challenging scenarios, preserving its multimodal behavior even during interactions with other agents.
4. Our experiments demonstrated that both IBC and DA-IBC are capable of producing multiple valid behaviors in complex urban environments, unlike BC, which consistently selects a single mode per scenario. Using a Bird’s Eye View (BEV) representation and the augmented training strategy, DA-IBC successfully learned multimodal policies, particularly evident at intersections where it was the only agent capable of selecting distinct driving strategies.

Overall, these findings highlight the potential of DA-IBC as a promising step toward autonomous driving policies that effectively balance reliability, safety, and multimodal decision-making. Future work should address conservative behaviors occasionally observed, improve the agent’s ability to resume motion after stops, and extend

evaluations to more complex and diverse environments. Additionally, it would be valuable to investigate a Guided IBC variant trained without explicit route information, but instead provided with external directional cues to indicate the desired path. As in many real-world autonomous driving scenarios, explicit route information may not always be available or reliable. Guided IBC approach would allow the agent to make informed decisions based on external directional cues, increasing flexibility in real-world environments.

REFERENCES

- ANTONELO, Eric *et al.* Investigating Behavior Cloning from Few Demonstrations for Autonomous Driving Based on Bird's-Eye View in Simulated Cities. *In: PROCEEDINGS of the Brazilian Conference on Intelligent Systems (BRACIS)*. [S.l.]: SBC, 2024. P. 155–168.
- ATTIA, Alexandre; DAYAN, Sharone. Global Overview of Imitation Learning. **arXiv preprint arXiv:1801.06503**, 2018.
- BALESNI, Mikita; TAMPUU, Ardi; MATIISEN, Tambet. **Controlling Steering with Energy-Based Models**. [S.l.: s.n.], 2023. arXiv: 2301.12264 [cs.R0]. Disponível em: <https://arxiv.org/abs/2301.12264>.
- BANSAL, Mayank; KRIZHEVSKY, Alex; OGALÉ, Abhijit. Chauffeurnet: Learning to Drive by Imitating the Best and Synthesizing the Worst. **arXiv preprint arXiv:1812.03079**, 2018.
- BENGIO, Yoshua; GOODFELLOW, Ian; COURVILLE, Aaron. **Deep Learning**. [S.l.]: MIT Press, 2017.
- BISHOP, Christopher M. **Pattern Recognition and Machine Learning**. New York: Springer, 2006.
- BOJARSKI, Mariusz *et al.* End to End Learning for Self-Driving Cars. **arXiv preprint arXiv:1604.07316**, 2016.
- CHAI, Yuning *et al.* Multipath: Multiple Probabilistic Anchor Trajectory Hypotheses for Behavior Prediction. **arXiv preprint arXiv:1910.05449**, 2019.
- CHEN, Dian *et al.* **Learning by Cheating**. [S.l.: s.n.], 2019. arXiv: 1912.12294 [cs.R0].
- CODEVILLA, Felipe *et al.* End-to-End Driving via Conditional Imitation Learning. *In: IEEE International Conference on Robotics and Automation (ICRA)*. [S.l.: s.n.], 2018.
- COUTO, Gustavo Claudio Karl; ANTONELO, Eric Aislan. Hierarchical Generative Adversarial Imitation Learning With Mid-Level Input Generation for Autonomous Driving on Urban Environments. **IEEE Transactions on Intelligent Vehicles**, v. 10, n. 3, p. 1841–1854, 2025. DOI: 10.1109/TIV.2024.3436587.
- FLORENCE, Pete *et al.* Implicit Behavioral Cloning. *In: PROCEEDINGS of the Conference on Robot Learning (CoRL)*. [S.l.]: PMLR, 2022. P. 158–168.
- FRANÇOIS-LAVET, Vincent *et al.* An Introduction to Deep Reinforcement Learning. **Foundations and Trends in Machine Learning**, v. 11, n. 3–4, p. 219–354, 2018.

GELMAN, Andrew *et al.* **Bayesian Data Analysis**. [S.l.]: Chapman and Hall/CRC, 1995.

HINTON, Geoffrey E. Training Products of Experts by Minimizing Contrastive Divergence. **Neural Computation**, v. 14, n. 8, p. 1771–1800, 2002.

HO, Jonathan; ERMON, Stefano. Generative Adversarial Imitation Learning. *In*: ADVANCES in Neural Information Processing Systems (NeurIPS). [S.l.: s.n.], 2016. P. 4565–4573.

KINGMA, Diederik P.; BA, Jimmy. Adam: A Method for Stochastic Optimization. **arXiv preprint arXiv:1412.6980**, 2014.

LECUN, Yann; BOTTOU, Léon, *et al.* Gradient-Based Learning Applied to Document Recognition. **Proceedings of the IEEE**, v. 86, n. 11, p. 2278–2324, 1998.

LECUN, Yann; CHOPRA, Sumit, *et al.* A Tutorial on Energy-Based Learning. **Predicting Structured Data**, 2006.

LI, Yunzhu; SONG, Jiaming; ERMON, Stefano. **InfoGAIL: Interpretable Imitation Learning from Visual Demonstrations**. [S.l.: s.n.], 2017. arXiv: 1703.08840 [cs.LG]. Disponível em: <https://arxiv.org/abs/1703.08840>.

LI, Yuxi. Deep Reinforcement Learning: An Overview. **arXiv preprint arXiv:1701.07274**, 2017.

LIAO, Bencheng *et al.* **DiffusionDrive: Truncated Diffusion Model for End-to-End Autonomous Driving**. [S.l.: s.n.], 2024. arXiv: 2411.15139 [cs.CV]. Disponível em: <https://arxiv.org/abs/2411.15139>.

MACHADO, Bruno M.; ANTONELO, Eric A. Investigating Diffusion Models for Offline Behavior Cloning in Autonomous Driving Scenarios. *In*: PROCEEDINGS of the Brazilian Conference on Robotics (CROS). [S.l.: s.n.], 2025. “Investigating Diffusion Models for Offline Behavior Cloning in Autonomous Driving Scenarios”, CROS 2025. Disponível em: https://www.researchgate.net/publication/393533806_Investigating_Diffusion_Models_for_Offline_Behavior_Cloning_in_Autonomous_Driving_Scenarios.

O’SHEA, Keiron. An Introduction to Convolutional Neural Networks. **arXiv preprint arXiv:1511.08458**, 2015.

PETRAZZINI, Irving GB; ANTONELO, Eric A. Proximal Policy Optimization with Continuous Bounded Action Space via the Beta Distribution. *In*: IEEE. 2021 IEEE Symposium Series on Computational Intelligence (SSCI). [S.l.: s.n.], 2021. P. 1–8.

POMERLEAU, Dean A. ALVINN: An Autonomous Land Vehicle in a Neural Network. *In: ADVANCES in Neural Information Processing Systems*. [S.l.: s.n.], 1989.

RHINEHART, Nicholas; MCALLISTER, Rowan; LEVINE, Sergey. Deep Imitative Models for Flexible Inference, Planning, and Control. **arXiv preprint arXiv:1810.06544**, 2018.

ROSS, S. *et al.* A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning. *In: PROCEEDINGS of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*. [S.l.: s.n.], 2011.

SANTOS, Felipe; ANTONELO, Eric; COUTO, Gustavo. An Initial Study of Bird's-Eye View Generation for Autonomous Vehicles using Cross-View Transformers. *In: ANAIS do XXII Encontro Nacional de Inteligência Artificial e Computacional*. Fortaleza/CE: SBC, 2025. P. 855–866. DOI: 10.5753/eniac.2025.14251. Disponível em: <https://sol.sbc.org.br/index.php/eniac/article/view/38774>.

SONG, Yang; ERMON, Stefano. Score-based generative modeling through stochastic differential equations. *In: INTERNATIONAL Conference on Learning Representations (ICLR)*. [S.l.: s.n.], 2021.

SUTTON, Richard S.; BARTO, Andrew G. **Reinforcement Learning: An Introduction**. 2. ed. [S.l.]: MIT Press, 2018.

WELLING, Max; TEH, Yee Whye. Bayesian learning via stochastic gradient Langevin dynamics. *In: OMNIPRESS. PROCEEDINGS of the 28th International Conference on Machine Learning (ICML)*. [S.l.: s.n.], 2011. P. 681–688.

YILDIRIM, Mustafa *et al.* **Behavioral Cloning Models Reality Check for Autonomous Driving**. [S.l.: s.n.], 2024. arXiv: 2409.07218 [cs.R0]. Disponível em: <https://arxiv.org/abs/2409.07218>.

ZHANG, Z. *et al.* Using Implicit Behavior Cloning and Dynamic Movement Primitive to Facilitate Reinforcement Learning for Robot Motion Planning. **IEEE Transactions on Robotics**, 2024. See full text for experiments and application notes. DOI: 10.1109/TR0.2024.3468770.

ZHANG, Zhejun *et al.* **End-to-End Urban Driving by Imitating a Reinforcement Learning Coach**. [S.l.: s.n.], 2021. arXiv: 2108.08265 [cs.CV]. Disponível em: <https://arxiv.org/abs/2108.08265>.

ZHENG, Yinan *et al.* Diffusion-Based Planning for Autonomous Driving with Flexible Guidance. **arXiv preprint arXiv:2501.15564**, 2025. Disponível em: <https://arxiv.org/abs/2501.15564>.

ZIEBART, Brian D. *et al.* Maximum Entropy Inverse Reinforcement Learning. *In: PROCEEDINGS of the AAAI Conference on Artificial Intelligence*. [S.l.: s.n.], 2008.