Agente VAMOS! Planejamento de Rotas Veiculares Cientes de Contexto Semântico com Agentes de LLM

Carnot Braun[1], Daniel L. Guidoni[2], Eduardo Cerqueira[3], Joahannes B. D. da Costa[4], Leandro Villas[1], Allan M. de Souza[1]

[1]Instituto de Computação - Universidade Estadual de Campinas - UNICAMP
[2]Universidade Federal de Ouro Preto - UFOP
[3]Universidade Federal do Pará - UFPA
[4]Universidade Federal de São Paulo - UNIFESP

c255785@dac.unicamp.br, guidoni@ufop.edu.br, cerqueira@ufpa.br
joahannes.costa@unifesp.br, {lvillas, allanms}@unicamp.br

Abstract. *Traditional navigation systems prioritize metric efficiency, such as time or distance, but often fail to interpret complex, context-dependent human intentions. Although Large Language Models (LLMs) demonstrate the potential to bridge this semantic gap, their direct integration into Intelligent Transportation Systems (ITS) faces critical barriers regarding scalability, latency, and connectivity dependence. To overcome these challenges, this work presents VAMOS(Vehicular Agent for Multi-objective Optimization and Semantics), a hybrid agent designed for efficient onboard operation. VAMOS decouples semantic reasoning from spatial optimization, combining local Small Language Models (SLMs) for intent interpretation with classic graph algorithms for route execution. Experimental evaluation across three urban scenarios demonstrates that VAMOS achieves accuracy and completeness exceeding 91% using compact models. Furthermore, the results highlight a favorable trade-off: although massive models show a marginal quality gain (3%), VAMOS offers a significant reduction in computational and communication overhead, validating the feasibility of semantically aware navigation assistants.*

Resumo. *Sistemas de navegação tradicionais priorizam a eficiência métrica, como tempo ou distância, mas falham frequentemente na interpretação de intenções humanas complexas e dependentes do contexto. Embora os Grandes Modelos de Linguagem (LLMs) demonstrem potencial para preencher essa lacuna semântica, sua integração direta em Sistemas de Transporte Inteligentes (ITS) enfrenta barreiras críticas de escalabilidade, latência e dependência de conectividade. Para superar esses desafios, este trabalho apresenta o VAMOS(Vehicular Agent for Multi-objective Optimization and Semantics), um agente híbrido desenhado para operar eficientemente embarcado. O VAMOS desacopla o raciocínio semântico da otimização espacial, combinando Pequenos Modelos de Linguagem (SLMs) locais para a interpretação de intenções com algoritmos de grafos clássicos para a execução de rotas. A avaliação experimental em três cenários urbanos demonstra que o VAMOS atinge acurácia e completude superiores a 91% utilizando modelos compactos. Além disso, os resultados evidenciam um trade-off favorável: embora modelos massivos apresentem um ganho marginal de qualidade (3%), o VAMOS oferece uma redução significativa no overhead computacional e de comunicação, validando a viabilidade de assistentes de navegação semanticamente conscientes.*

1. Introdução

Com o avanço da Inteligência Artificial em ambientes urbanos, o problema de recomendação de rotas veiculares evolui da simples escolha do caminho mais rápido entre origem e destino para a definição de trajetos personalizados, orientados por diferentes contextos e requisitos do usuário [Barbon et al. 2025, De Souza et al. 2017, de Souza et al. 2019]. Esses requisitos podem variar dinamicamente conforme preocupações com segurança, condições ambientais, estado do veículo ou preferências pessoais ao longo do trajeto [Wang et al. 2022, Zhang et al. 2024b, Luca et al. 2021, Paiva et al. 2021]. Nesse cenário, a personalização de rotas não apenas melhora a experiência individual de navegação, mas também contribui para uma gestão de tráfego mais eficiente, com potencial redução de congestionamentos e emissões de poluentes [Benmessaoud et al. 2023].

Atualmente, grande parte das informações de rotas veiculares são contextuais e expressas de forma descritiva, ambígua e dependente de situação, como solicitações não assertivas relacionadas a necessidades veiculares (por exemplo, meu carro está ficando sem combustível, preciso abastecer). Esse tipo de informação é difícil de modelar por abordagens algorítmicas clássicas, que operam sobre dados estruturados e não são projetadas para interpretar intenções semânticas ou graus de necessidade expressos em linguagem natural. Métodos baseados apenas na extração de palavras-chave ou regras fixas tendem a falhar ao distinguir entre requisitos obrigatórios, opções desejáveis e preferências contextuais [Jiang et al. 2025].

Modelos de Linguagem de Grande Porte (*Large Language Model (LLM)*) emergem como uma alternativa promissora para interpretar intenções humanas expressas em linguagem natural, considerando contexto, ambiguidade e prioridades implícitas. Contudo, o uso direto de LLM para resolver problemas de roteamento apresenta limitações importantes, como falta de garantias de optimalidade, alto custo computacional e dificuldades de integração com sistemas de navegação reais. Uma alternativa natural consiste na adoção de agentes híbridos, nos quais o raciocínio semântico é desacoplado da execução algorítmica, permitindo que cada componente atue em sua especialidade [Chen et al. 2024a, Braun et al. 2025].

Trabalhos recentes demonstram o potencial de LLM e de *frameworks* baseados em agentes para diversas aplicações no domínio urbano [Wu et al. 2023, Xi et al. 2023]. As aplicações variam desde a construção de grafos de conhecimento urbanos e a simulação de planejamento participativo multiagente até a análise de dados de mobilidade para compreender intenções de visita e a geração de trajetórias pessoais de mobilidade. Tais abordagens abrem novas possibilidades para a análise e o planejamento em cidades inteligentes [Gong et al. 2024, Wang et al. 2024, Ning and Liu 2024]. No entanto, nesses trabalhos, observa-se a dificuldade de adaptar agentes às diferentes características dos cenários veiculares, especialmente ao tentar ajustar-se a distintos tipos de Points of Interest (POI). Como os dados de mobilidade veicular (rotas, destinos e POIs associados a serviços veiculares) estão intrinsecamente ligados a características regionais específicas, torna-se desafiador aplicar um modelo treinado em outro domínio sem adaptações significativas.

Outras soluções [Qin et al. 2025, Huang et al. 2024, Wang et al. 2023] exploram as capacidades de raciocínio dos LLM para prever o próximo destino de um indivíduo no transporte público, resolver problemas de roteamento veicular a partir de descrições em linguagem natural e identificar padrões gerais de mobilidade humana. Esses métodos transformam dados e problemas espaço-temporais em representações que podem ser processadas por LLM, possibilitando soluções mais flexíveis. Contudo, tais previsões

baseiam-se majoritariamente em históricos de visita e podem não refletir de forma precisa os desejos ou preferências atuais do usuário. Isso evidencia a necessidade de métodos que considerem simultaneamente o contexto do usuário e o cenário veicular.

Embora essas abordagens demonstrem o potencial dos LLM para lidar com informações semânticas, muitas dependem fortemente de modelos centralizados ou de históricos de mobilidade, o que dificulta a adaptação a cenários veiculares dinâmicos e a diferentes distribuições espaciais de POI. Além disso, soluções centralizadas tendem a enfrentar desafios de escalabilidade, latência e privacidade, especialmente em cenários de mobilidade urbana em larga escala [Zhang et al. 2024a, de Souza et al. 2022]. Isso motiva a investigação de abordagens com tomada de decisão local, nas quais modelos menores (*Small Language Models* – SLMs) podem operar próximos à origem dos dados, como no próprio veículo ou em infraestrutura de borda, reduzindo dependência de comunicação constante com servidores centrais, visto que o custo de processamento e infererência é bem menor [Irugalbandara et al. 2024].

Diante dessas limitações, este artigo propõe o VAMOS (*Vehicular Agent for Multi-objective Optimization and Semantics*), um agente híbrido para recomendação e adaptação de rotas veiculares. O VAMOS combina um módulo cognitivo baseado em *Small Language Model (SLM)*, responsável por interpretar intenções, requisitos e preferências do usuário, com ferramentas algorítmicas clássicas de roteamento, responsáveis pela geração e modificação eficiente das rotas. Essa separação permite explorar a eficiência e a optimalidade de algoritmos de grafos, enquanto o LLM atua exclusivamente no raciocínio semântico e contextual, coordenando as ações do agente.

Em resumo, as contribuições deste trabalho são: i) proposta e o desenvolvimento do VAMOS, um agente híbrido para planejamento e seleção de rotas veiculares, capaz de interpretar intenções semânticas do usuário e acionar ferramentas de roteamento para adaptação de trajetos de forma contextualizada; ii) avaliação abrangente do VAMOS em três cidades com características urbanas distintas, considerando diferentes tipos de intenções e requisitos de mobilidade, de modo a analisar a robustez e a generalização do agente; e iii) disponibilização da implementação do VAMOS e dos *prompts* utilizados nos experimentos[1], visando facilitar a reprodutibilidade e a ciência aberta.

O restante deste artigo é organizado da seguinte forma: A Seção 2 apresenta os trabalhos relacionados. A Seção 3 descreve a proposta VAMOS. A avaliação de desempenho e os resultados são discutido na Seção 4. Por fim, a Seção 5 apresenta as conclusões e os trabalhos futuros.

2. Trabalhos Relacionados

A literatura recente tem investigado extensivamente o uso de LLM em sistemas de transporte inteligentes e aplicações de computação urbana. Embora essas abordagens demonstrem avanços expressivos em raciocínio semântico e capacidade de generalização, grande parte delas enfrenta limitações críticas do ponto de vista de sistemas, especialmente no que se refere à latência de inferência, eficiência computacional e custo de operação. Esses fatores tornam sua adoção problemática em cenários veiculares, nos quais decisões devem ser tomadas em tempo quase real e frequentemente em dispositivos com recursos limitados.

Do ponto de vista arquitetural, observa-se que muitas soluções baseadas em LLM assumem infraestruturas centralizadas e dependem de ciclos iterativos de inferência ou de chamadas frequentes a serviços externos. No domínio da gestão urbana, Ning *et*

[1] `https://github.com/carnotbraun/VAMOS`

al. [Ning and Liu 2024] propõem o UrbanKGent, uma arquitetura voltada à construção de grafos de conhecimento urbanos por meio do ajuste fino de modelos Llama-2 e Llama-3, combinados com ferramentas geoespaciais externas. Apesar de reduzir o esforço humano, o método depende de múltiplas iterações e acessos externos, o que eleva significativamente a latência de processamento e dificulta sua aplicação em cenários dinâmicos ou distribuídos.

De forma semelhante, Zhou *et al.* [Zhou et al. 2024] introduzem uma abordagem de planejamento participativo baseada em simulações multiagente do tipo *fishbowl*, nas quais modelos como GPT-4 Vision e GPT-3.5 representam diferentes atores urbanos. Embora inovadora do ponto de vista metodológico, essa estratégia exige a execução simultânea de diversos agentes baseados em LLM, resultando em elevado consumo de *tokens*, alta latência e custos proibitivos para aplicações contínuas ou em larga escala.

No contexto de análise semântica e predição de mobilidade, busca-se superar as limitações de modelos puramente estatísticos. Gong *et al.* [Gong et al. 2024] propõem um *framework* que combina uma rede de memória de intenção (VIMN) com o modelo `TinyLlama-1B` ajustado via LoRA, priorizando eficiência de parâmetros. Embora essa abordagem reduza custos de treinamento, sua especialização em tarefas específicas limita a generalização para cenários veiculares mais complexos. Qin *et al.* [Qin et al. 2025], por sua vez, exploram o uso *zero-shot* do `ChatGPT-3.5` para predição de viagens, estruturando históricos de mobilidade em contextos hierárquicos. Contudo, a dependência de APIs proprietárias e a sensibilidade ao tamanho da janela de contexto impõem desafios relevantes de custo, privacidade e escalabilidade.

Outras linhas de pesquisa investigam o uso de LLM em problemas de otimização e geração de dados. Métodos como o *FWTRoutes* [Huang et al. 2024] utilizam LLMs para gerar código com auto-reflexão para resolver Problemas de Roteamento de Veículos (VRP). Apesar de promissores, tais métodos introduzem incertezas computacionais associadas à geração de código em tempo real, o que é particularmente crítico em aplicações veiculares. Abordagens voltadas à geração de dados sintéticos, como o Traj-LLM [Yang et al. 2025] e métodos baseados em RAG [Wang et al. 2024], operam majoritariamente em modo *offline* e dependem de curadoria textual ou acesso a APIs de alto custo, não sendo adequadas para tomada de decisão em tempo real.

No que diz respeito à navegação e coordenação de veículos, soluções como o CityNav [Zhou et al. 2025] e o LiMeDa [Chen et al. 2024b] adotam arquiteturas híbridas ou hierárquicas. O CityNav utiliza múltiplos agentes baseados em LLM para coordenação global e local, obtendo bons resultados em grandes redes, porém à custa de inferência contínua e elevada carga computacional. O LiMeDa combina um LLM para coordenação de tarefas com um módulo de Aprendizado por Reforço para navegação, reduzindo parcialmente o custo de inferência, mas herdando limitações de escalabilidade do módulo de RL e exigindo manutenção de grandes memórias de experiência.

A Tabela 1 sintetiza essas abordagens sob a perspectiva de arquitetura, tipo de modelo (LLM ou SLM), estratégia de eficiência e limitações sistêmicas. Em síntese, embora a literatura demonstre o potencial dos LLM para lidar com aspectos semânticos da mobilidade urbana, observa-se uma lacuna importante no que diz respeito ao uso desses modelos como assistentes cognitivos acoplados a sistemas de roteamento eficientes, especialmente em arquiteturas distribuídas e de baixa latência. Poucos trabalhos investigam de forma sistemática como decisões semânticas podem ser integradas a motores algorítmicos de roteamento sem incorrer nos elevados custos e atrasos associados a inferências centralizadas e repetitivas. Diferentemente de abordagens puramente centradas em LLMs, o VAMOS adota uma arquitetura híbrida e distribuída, na qual modelos de menor porte

Tabela 1. Comparação de abordagens baseadas em modelos de linguagem aplicadas à mobilidade urbana e navegação veicular, sob a perspectiva de arquitetura de sistema, tipo de modelo e escalabilidade.

Método	Domínio	Tipo de Modelo	Arquitetura	Modo de Operação	Escalabilidade	Limitação Principal
CityNav	Navegação Veicular	LLM	Hierárquica (Global/Local)	Online	Média	Alta latência devido à inferência contínua de múltiplos agentes
LiMeDa	Coordenação/ Navegação	LLM + RL	Híbrida Centralizada	Online	Média	Escalabilidade limitada pelo módulo de RL e memória de experiências
UrbanKGent	Gestão Urbana (KGC)	LLM	Centralizada com Ferramentas	Online / Iterativo	Baixa	Latência elevada devido a refinamentos sucessivos e chamadas externas
Zhou *et al.*	Planejamento Urbano	LLM	Multiagente Centralizado	Online (Simulação)	Baixa	Alto custo de tokens e inviabilidade em larga escala
Gong *et al.*	Análise de Mobilidade	SLM	Centralizada	Online	Média	Especialização excessiva e baixa generalização
LingoTrip	Predição de Viagens	LLM (API)	Centralizada	Online	Baixa	Dependência de API proprietária e custo elevado
Traj-LLM	Geração de Trajetórias	LLM	Centralizada	Offline	Alta	Não aplicável para decisão em tempo real
FWTRoutes	Roteamento (VRP)	LLM	Centralizada	Online	Baixa	Fragilidade e incertezas na geração de código em tempo real
VAMOS	Roteamento Veicular	**SLM**	**Distribuída/ Híbrida**	**Online**	**Alta**	Decisão semântica limitada ao escopo local do agente

(SLMs) realizam o raciocínio semântico localmente, enquanto algoritmos clássicos de grafos são responsáveis pela execução eficiente do roteamento. Essa separação reduz a carga computacional, minimiza a latência de decisão e torna a abordagem mais alinhada aos requisitos de sistemas veiculares distribuídos, característicos do domínio de redes e sistemas distribuídos.

3. VAMOS

3.1. Visão Geral do Sistema

O VAMOS é projetado como um sistema veicular modular e distribuído, estruturado em camadas funcionais que organizam a interação entre usuários, algoritmos de roteamento e mecanismos de decisão semântica. Diferentemente de soluções que centralizam o processo de planejamento em modelos de linguagem, o VAMOS adota uma separação explícita entre cálculo de rotas e tomada de decisão, permitindo interoperabilidade com soluções de roteamento já consolidados e integração flexível com arquiteturas de ITS.

A Figura 1 apresenta a arquitetura lógica do sistema, organizada em três camadas principais, cada uma responsável por um conjunto bem definido de funções. Essa organização favorece a extensibilidade do sistema e permite que o VAMOS seja integrado tanto a plataformas embarcadas quanto a infraestruturas veiculares distribuídas, sem introduzir dependências rígidas ou sobrecarga excessiva de comunicação. Sendo estas:

Camada de Entrada (A): responsável por capturar as informações fornecidas pelo usuário e pelo veículo conectado, incluindo requisições em linguagem natural, preferências persistentes (por exemplo, evitar determinadas regiões) e contexto local disponível no momento da solicitação.

Núcleo de Decisão (B): camada central do sistema, onde ocorre a geração de alternativas de rota e a decisão final de navegação. Essa camada concentra os principais módulos funcionais do VAMOS e pode ser executada localmente no veículo ou em infraestrutura de borda, permitindo operação com baixa dependência de conectividade.

Camada de Dados Externos (C): provê informações complementares ao processo decisório, como POI, dados de tráfego e condições climáticas. Essas informações podem ser mantidas em cache local e atualizadas sob demanda, não sendo requisito para o funcionamento contínuo do sistema.

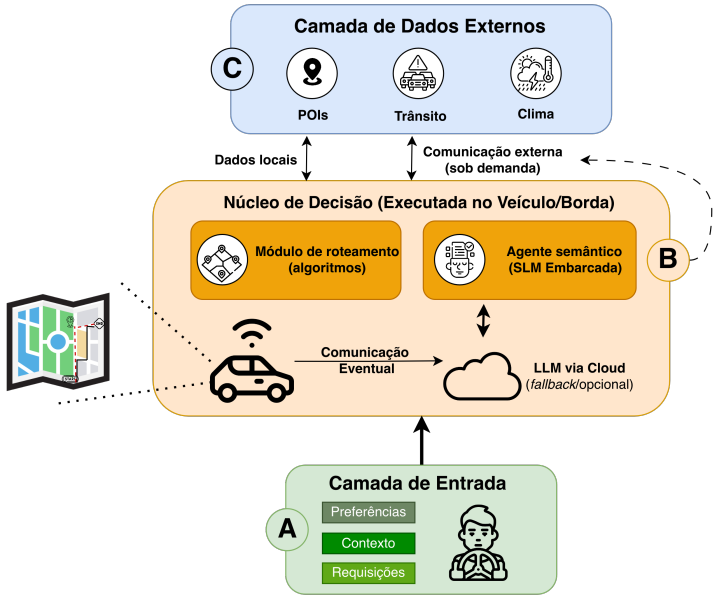


Figura 1. Arquitetura do sistema VAMOS. O sistema organiza-se em camadas funcionais, nas quais um agente semântico baseado em SLM atua como núcleo de decisão, operando sobre rotas calculadas por módulos clássicos de roteamento e dados contextuais externos.

A arquitetura proposta permite incorporar o VAMOS a diferentes aplicações veiculares. A atualização de dados externos ocorre de forma oportunista, evitando inferência contínua ou sincronização global, o que reduz o *overhead* de comunicação e favorece a execução distribuída em ambientes veiculares dinâmicos. Essa estratégia segue o princípio de comunicação oportunista em redes veiculares e sistemas ITS, em que a troca de informações com infraestruturas externas ocorre apenas quando há conectividade adequada, sem pressupor comunicação contínua ou garantida [Zheng et al. 2015]. Nesse modelo, dados como POI, tráfego ou clima são atualizados de forma assíncrona e armazenados localmente, enquanto o núcleo decisório permanece funcional mesmo com conectividade intermitente. Essa abordagem reduz o *overhead* de comunicação, aumenta a robustez em ambientes dinâmicos e é especialmente adequada a aplicações embarcadas em veículos conectados.

3.2. Módulos Funcionais do Núcleo de Decisão

O Núcleo de Decisão do VAMOS é composto por três módulos funcionais principais, cada um associado a uma responsabilidade específica no processo de recomendação. Embora apresentados separadamente, esses módulos operam de forma coordenada dentro da mesma camada, conforme ilustrado na Figura 1. A seguir, são apresentados os três módulos.

Módulo de Roteamento (RET): é responsável pelo cálculo de rotas candidatas sobre a malha viária urbana, modelada como um grafo $G = (V, E)$, em que V representa interseções e E representa vias. A partir de um par origem–destino (s, d), o módulo gera um conjunto de caminhos candidatos $\mathcal{P} = \{\rho_{direct}, \rho^1_{semantic}, \ldots, \rho^k_{semantic}\}$ utilizando algoritmos clássicos de grafos, como *Dijkstra* multiobjetivo. O módulo é independente do agente semântico, o que permite sua substituição por variantes mais eficientes ou especializadas sem impacto na lógica de decisão do sistema.

Módulo de Contextualização Geoespacial (GCT): é responsável por associar informações semânticas do ambiente urbano aos elementos do grafo G. Em particular,

ele identifica e indexa POI relevantes (por exemplo, hospitais, postos de combustível e supermercados), mapeando-os para nós e arestas da rede viária. Nos experimentos realizados, essa associação é construída a partir de dados do *OpenStreetMap*, mantidos em cache local. Esse módulo fornece ao sistema a capacidade de interpretar o significado funcional das rotas, indo além de métricas puramente geométricas.

Agente Semântico (CRAT): é responsável pela decisão final entre as rotas candidatas geradas pelo módulo de roteamento. Implementado sobre um SLM embarcado, o agente analisa as intenções do usuário, o contexto disponível e os metadados das rotas candidatas, aplicando uma hierarquia de decisão explícita. Essa hierarquia prioriza requisitos críticos (por exemplo, urgência) sobre critérios de conveniência ou eficiência. Opcionalmente, o agente pode consultar um LLM via nuvem como mecanismo de apoio, porém essa comunicação é eventual e não constitui dependência estrutural do sistema.

Essa separação explícita entre geração de alternativas e decisão semântica garante previsibilidade, eficiência computacional e robustez operacional. Ao concentrar o raciocínio contextual em um agente leve e embarcado, o VAMOS evita os custos e a latência associados a arquiteturas centralizadas, mantendo compatibilidade com sistemas de navegação já consolidados.

3.3. Fluxo Operacional

O fluxo operacional do VAMOS segue uma abordagem de *Reason-over-Routes*, na qual o sistema raciocina sobre um conjunto reduzido de rotas candidatas, em vez de gerar trajetos diretamente por meio do modelo de linguagem, conforme ilustrado na Figura 2. Inicialmente, o motor de roteamento calcula rotas candidatas entre a origem s e o destino d, incluindo uma rota base otimizada para eficiência e rotas alternativas que incorporam paradas em POI relevantes.

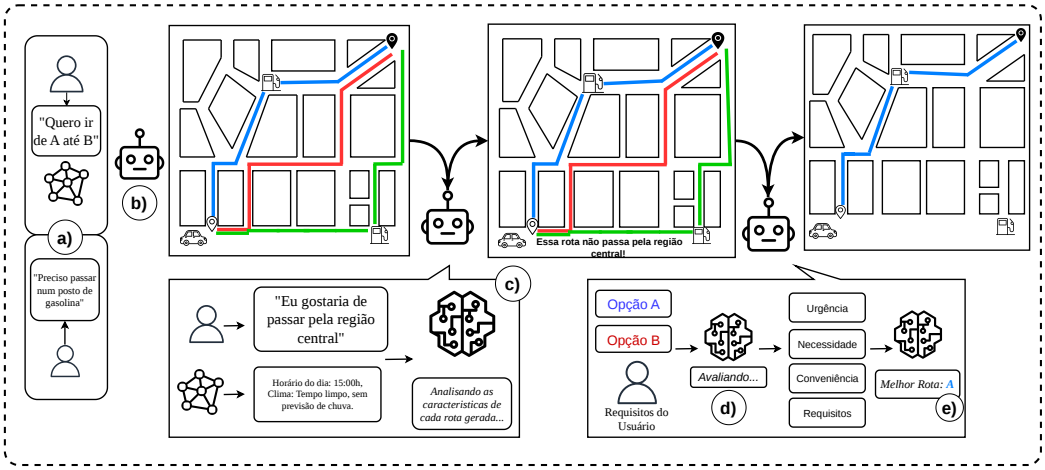


Figura 2. Fluxo operacional do VAMOS. O sistema adota uma abordagem de *Reason-over-Routes*, na qual o agente avalia um conjunto reduzido de rotas candidatas enriquecidas com contexto, selecionando a alternativa que melhor atende à intenção e aos requisitos do usuário.

Em paralelo, o agente semântico analisa a solicitação do usuário, identificando tarefas, requisitos e preferências implícitas. As rotas candidatas são então enriquecidas com metadados contextuais, como tempo estimado, custo marginal de desvios e tarefas potencialmente concluídas. Com base nesse cenário estruturado, o agente aplica uma hierarquia de decisão que prioriza necessidades urgentes sobre conveniência e eficiência,

selecionando a rota que maximiza a utilidade para o usuário. A saída final consiste na rota escolhida, acompanhada de uma justificativa explícita, permitindo interpretabilidade e integração direta com sistemas de navegação. Assim, o pipeline de decisão do VAMOS é estruturado em cinco etapas sequenciais:

1. **Interpretação de Intenção**: o agente analisa a requisição do usuário em linguagem natural e a decompõe em tarefas atômicas, atribuindo níveis de importância e categorias semânticas.
2. **Geração de Rotas Candidatas**: o módulo de roteamento calcula rotas diretas e alternativas que incorporam POI relevantes.
3. **Enriquecimento Contextual**: as rotas são anotadas com informações de tempo, custo de desvio e tarefas concluídas.
4. **Avaliação Hierárquica**: o agente aplica uma hierarquia de decisão rígida, priorizando urgência, necessidade, conveniência e eficiência.
5. **Ação Final**: o sistema retorna a rota selecionada, a ação requerida (ex.: adicionar parada intermediária) e uma justificativa explicável ao usuário.

Por fim, o agente semântico do VAMOS emprega estratégias de raciocínio estruturado para garantir consistência e previsibilidade nas decisões. Técnicas inspiradas em *Chain-of-Thought* [Wei et al. 2022] são utilizadas internamente no agente para estruturar o raciocínio e para decompor solicitações complexas e reduzir ambiguidades semânticas. Diferentemente de abordagens baseadas em geração livre, o agente opera exclusivamente sobre representações estruturadas, assegurando saídas determinísticas e integráveis ao sistema de navegação.

4. Avaliação de Desempenho

O objetivo desta avaliação é verificar se o VAMOS consegue (i) *interpretar* corretamente a intenção do usuário em linguagem natural e (ii) *operacionalizar* essa intenção em uma decisão de navegação (seleção e/ou edição de rota) consistente com a hierarquia de decisão do agente. Diferente de avaliar o modelo de linguagem isoladamente, consideramos o VAMOS como um sistema completo: o resultado final depende da geração de rotas candidatas pelo módulo de roteamento e da decisão do agente sobre essas alternativas. Essa escolha é coerente com a proposta do VAMOS como assistente híbrido, no qual algoritmos clássicos resolvem a otimização espacial e o SLM/LLM atua como camada de decisão semântica e contextual.

4.1. Métricas

A avaliação de desempenho foi realizada com base em três métricas principais, que fornecem uma visão holística da eficiência e viabilidade da solução proposta.

Acurácia (*Accuracy*): Proporção de execuções em que o agente seleciona a alternativa compatível com a política do VAMOS. Consideramos correto quando o sistema escolhe $\rho_{semantic}$ nos casos em que a intenção exige a inclusão de um POI (p. ex., combustível/hospital) e escolhe ρ_{direct} quando o desvio não é justificado pela política.

Completude (*Completeness*): Proporção de execuções em que o agente identifica corretamente o tipo de ação requerida para satisfazer a intenção, isto é, sinalizar a necessidade de adicionar um ponto no trajeto quando a solicitação implica visitar um POI, ou sinalizar que nenhuma modificação de rota é necessária. Essa métrica captura entendimento de intenção com menor sensibilidade a variações geométricas entre rotas candidatas (por exemplo, quando duas alternativas possuem custos muito próximos).

Tabela 2. Análise de desempenho de VAMOS em todos os cenários com diferentes modelos, usando métricas de precisão e completude (em negrito, os melhores entre eles)

Modelos	Cenários					
	São Paulo		Salvador		Belém	
	Acc. (%)	Complet. (%)	Acc. (%)	Complet. (%)	Acc. (%)	Complet. (%)
LFM2.5 1.2B	61.10	66.67	62.20	66.40	69.01	70.60
Phi-3.5 3.82B	42.86	64.29	64.71	73.53	53.33	83.33
Qwen3 4B	91.49	92.01	**92.86**	**93.40**	93.45	94.33
Mistral 7B	48.78	58.54	53.19	59.57	54.35	65.22
Llama-3 8B	88.24	90.20	84.85	91.44	91.94	92.62
gpt-3.5-turbo	**96.72**	**97.22**	86.11	87.67	**95.83**	**96.22**

Overhead fim-a-fim (*End-to-End Overhead*): Reportamos o tempo médio por execução como uma aproximação do custo fim-a-fim do ciclo de recomendação do sistema (consulta geoespacial, geração das rotas candidatas e decisão do agente). Para modelos locais, esse *overhead* corresponde ao tempo total observado no nó embarcado/edge. Para o modelo via API, esse *overhead* inclui também o tempo de comunicação com a nuvem medido no cliente, ou seja, a ida e volta do pedido.

4.2. Cenários e cidades

A avaliação foi conduzida em três cidades brasileiras: São Paulo, Salvador e Belém. Elas foram escolhidas por apresentarem características urbanas distintas, com diferenças de topologia viária e distribuição de POI, o que permite avaliar a generalização do VAMOS fora de um único mapa.

Mantivemos duas categorias de cenário, pois representam intenções frequentes em navegação e expõem o comportamento de decisão do agente em situações contrastantes: **(i) Urgência** (p. ex., posto de combustível/hospital), em que a política do VAMOS tende a aceitar desvios para cumprir requisitos; e **(ii) Conveniência** (p. ex., mercado/parque), em que o desvio pode ser rejeitado quando o custo contextual é alto. Em cada instância, o sistema calcula um conjunto de rotas candidatas contendo uma rota direta ρ_{direct} e uma ou mais rotas alternativas $\rho_{semantic}$ (incorporando POI inferidos a partir das tarefas, preferências do usuário e contexto do cenário). O agente recebe esse conjunto e retorna a rota escolhida e a ação requerida.

4.3. Resultados

A Tabela 2 apresenta Acurácia e Completude para seis modelos: cinco modelos locais (*LFM2.5-1.2B*, *Qwen3-4B*, *Phi-3.5-3.82B*, *Mistral-7B* e *Llama-3-8B*) e um baseline via API (*gpt-3.5-turbo*). O **Qwen3-4B** foi o modelo local mais consistente nas três cidades, com Acurácia entre 91.49–93.45% e Completude entre 92.01–94.33%, o que indica que o agente, na maioria dos casos, (i) reconhece corretamente quando é necessário inserir um POI e (ii) seleciona a alternativa coerente com a hierarquia de decisão do sistema, este mesmo que demonstra o modelo *baseline*, no qual, foi desenvolvida toda a aplicação.

O **gpt-3.5-turbo** obteve os melhores resultados globais em São Paulo e Belém (Acurácia de 96.72% e 95.83%, Completude de 97.22% e 96.22%, respectivamente), sugerindo que modelos hospedados via API podem reduzir erros de decisão quando as rotas candidatas são próximas em custo. Em Salvador, porém, o desempenho do *gpt-3.5-turbo* (86.11% de Acurácia e 87.67% de Completude) ficou abaixo do Qwen3-4B (92.86%

Tabela 3. Análise do *overhead* fim-a-fim do VAMOS em todos os cenários com diferentes modelos, para cada um dos dois principais cenários modelados para avaliação (em negrito, os melhores entre eles).

Modelos	Cenários					
	São Paulo		Salvador		Belém	
	Urgência	Conveniência	Urgência	Conveniência	Urgência	Conveniência
LFM2.5 1.2B	188.73s	166.26s	98.63s	129.15s	**97.43s**	**111.1s**
Phi-3.5 3.82B	100.77s	81.55s	56.70s	**53.13s**	**44.37s**	55.58s
Qwen3 4B	127.27s	118.67s	**95.63s**	83.18s	**54.39s**	**51.21s**
Mistral 7B	156.14s	187.66s	206.88s	256.77s	**79.77s**	**50.41s**
Llama-3 8B	156.59s	170.8s	**84.66s**	**86.03s**	215.34s	222.34s
gpt-3.5-turbo	**250.76s**	181.4s	80,15s	**69,39s**	**63.72s**	76.39s

e 93.40%), o que indica que o ganho de qualidade do modelo via API não é uniforme e depende do cenário urbano e das instâncias avaliadas.

Os modelos **Phi-3.5** e **Mistral-7B** apresentaram acurácia baixa em todas as cidades (p. ex., 42.86% e 48.78% em São Paulo), mas com completude relativamente superior em comparação à acurácia (p. ex., 64.29% e 58.54% em São Paulo). Essa diferença sugere um padrão específico de falha: em parte das execuções o agente identifica corretamente que há (ou não há) necessidade de ação, porém erra ao escolher entre alternativas candidatas quando elas têm custos próximos ou quando a justificativa textual do modelo entra em conflito com o identificador de rota selecionado. Na prática, isso aponta que a principal fonte de erro não está apenas no entendimento da intenção, mas na etapa de comparação e seleção entre rotas candidatas.

A Tabela 3 apresenta o *overhead* fim-a-fim médio por categoria. Há variações por cidade e por modelo. Em cenários locais, o **Qwen3-4B** apresenta *overhead* menor em Belém (54.39s/51.21s) do que em São Paulo (127.27s/118.67s), o que é compatível com diferenças na densidade do grafo, número de POI consultados e custo de geração das rotas candidatas.

Ao comparar execução local *versus* API, é importante separar dois termos. Para modelos locais, o *overhead* medido é dominado pelo processamento no próprio nó (consulta + roteamento + inferência local). Para a API, além do processamento, existe um componente de rede, definido conforme Equação (1).

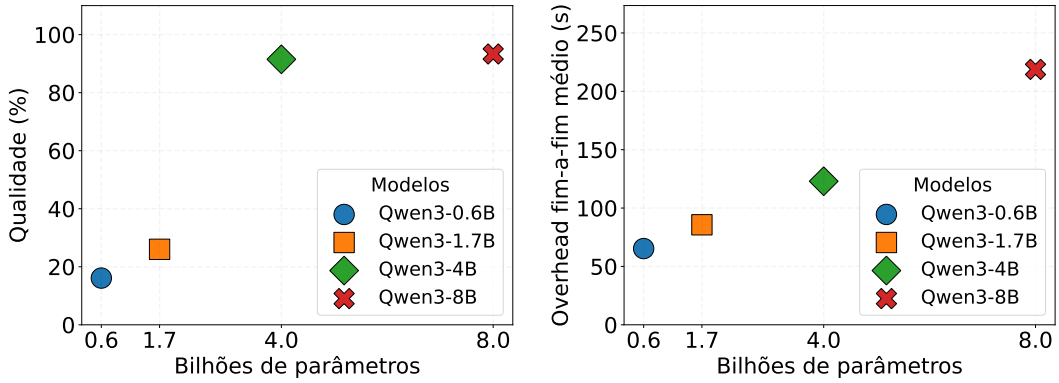
$$T_{\text{API}} \approx T_{\text{proc}} + T_{\text{net}} \tag{1}$$

Onde T_{net} inclui latência de ida e volta, variação (*jitter*) e eventuais retransmissões. Mesmo quando o tempo total medido da API é competitivo em alguns cenários (por exemplo, Salvador: 80.15s/69.39s), esse termo T_{net} torna o sistema dependente de conectividade e sujeito a degradações abruptas em mobilidade. Na prática, isso é relevante em cenários veiculares: a disponibilidade da rede muda com cobertura, congestionamento e *handover*, e o custo de comunicação não é controlável pelo sistema.

Esse ponto motiva a escolha do VAMOS por SLMs embarcadas e cache local: ao manter grafo viário e POI em cache e executar a decisão semântica localmente, o sistema reduz dependência de conexão contínua para operar sua lógica de personalização. Essa configuração contrasta com navegadores amplamente utilizados (por exemplo, Waze/Google Maps), que dependem de conexão frequente para atualização e recomputação contínua. No VAMOS, a decisão de inserir/evitar POI e a seleção entre alternativas ocorre

com dados locais, e fontes externas (quando disponíveis) entram como enriquecimento opcional, não como pré-requisito para funcionamento.

Além da comparação entre diferentes modelos, investigou-se o impacto do tamanho do modelo sobre dois aspectos centrais do VAMOS: i) a qualidade da decisão semântica e ii) o *overhead* computacional fim-a-fim. Essa análise é fundamental para cenários veiculares e de borda, onde recursos computacionais, consumo energético e latência são restritos.



(a) Qualidade *versus* Tamanho. (b) *Overhead versus* Tamanho.

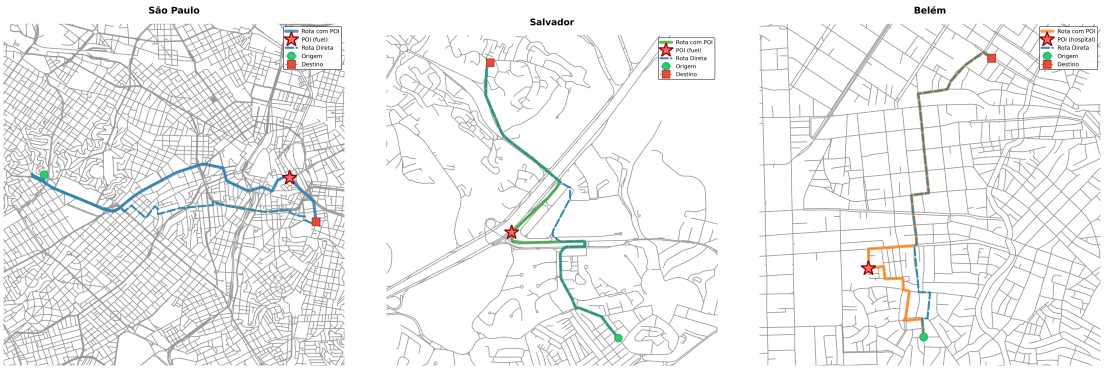
Figura 3. *Tradeoff* entre os modelos de diferentes tamanhos.

A Figura 3(a) apresenta o *trade-off* entre qualidade de decisão (Precisão) e tamanho do modelo para variantes da mesma família. Observa-se um ganho expressivo de qualidade ao aumentar o modelo de versões menores até aproximadamente 4B parâmetros. A partir desse ponto, o ganho marginal torna-se reduzido, indicando um ponto de inflexão no qual o aumento de complexidade do modelo não se traduz em melhoria proporcional de desempenho, considerando esse cenário. Esse comportamento sugere que, para a tarefa de decisão semântica sobre rotas candidatas, modelos de porte intermediário são suficientes para capturar as relações contextuais necessárias, enquanto modelos maiores introduzem custo adicional sem benefício equivalente.

A Figura 3(b) ilustra o crescimento do *overhead* de comunicação em função do tamanho do modelo. Diferentemente da curva de qualidade, o *overhead* apresenta crescimento monotônico, refletindo maior custo de inferência, uso de memória e tempo de processamento à medida que o número de parâmetros aumenta. Em cenários embarcados ou de borda, esse comportamento é particularmente crítico, pois impacta diretamente a viabilidade de execução local do agente e o tempo de resposta percebido pelo usuário.

A análise conjunta desses resultados evidencia um trade-off estrutural: enquanto modelos maiores podem reduzir erros pontuais de decisão, o custo computacional cresce de forma desproporcional. Nesse contexto, o Qwen3-4B emerge como um ponto de equilíbrio entre qualidade e *overhead*, oferecendo desempenho comparável a modelos maiores, porém com custo compatível com execução em borda.

Tais resultados quantitativos indicam que a escolha do modelo impacta diretamente não apenas a qualidade da decisão, mas também a viabilidade operacional do sistema em cenários reais. As Figuras 4(a) a 4(c) ilustram qualitativamente como essas decisões se materializam em rotas com paradas intermediárias nos diferentes contextos urbanos avaliados.



(a) Desvio Urgência (b) Desvio com requisitos e preferências (c) Desvio de requisitos

Figura 4. Exemplo de como o VAMOS compreende o que o usuário deseja, extrai a intenção e os requisitos, e assim escolhe o POI desejado e o melhor cenário para a situação.

Por fim, os resultados indicam que o VAMOS generaliza de forma consistente a política de seleção/edição para diferentes cidades, desde que o modelo subjacente se mantenha estável. Ademais, a complexidade intrínseca de cada cidade impacta de maneira significativa a acurácia, sobretudo porque distintas áreas urbanas podem apresentar níveis heterogêneos de complexidade e, simultaneamente, POI estruturalmente muito semelhantes em determinadas zonas, o que aumenta o risco de ambiguidades na seleção.

Adicionalmente, observa-se que, embora alguns modelos de maior porte, inclusive aqueles acessados via API, tenham alcançado acurácias superiores a 95%, as Slms, em especial o modelo *baseline*, apresentaram desempenho quantitativamente muito próximo. Este resultado é particularmente relevante ao se considerar que, no caso de *LLMs*, sua execução em dispositivos de borda implicaria custos adicionais substanciais, tanto em termos de recursos computacionais quanto de latência e consumo de energia.

Por fim, em comparação com sistemas amplamente utilizados, como Waze e Google Maps, que requerem conectividade contínua à Internet para realizar ajustes dinâmicos de rota e outras atualizações em tempo real, o VAMOS distingue-se por manter tais informações em localmente. Essa característica reduz a dependência de conectividade, potencialmente diminuindo a latência e aumentando a robustez operacional em cenários com acesso limitado à rede.

5. Conclusão

O artigo apresenta o VAMOS, um sistema híbrido de recomendação e adaptação de rotas que combina algoritmos de grafos com uma camada de decisão baseada em modelos de linguagem. A arquitetura separa a geração de rotas da decisão final, permitindo traduzir intenções em linguagem natural em ações estruturadas e explicáveis. Nos experimentos em três cidades brasileiras, o desempenho mostrou forte dependência da estabilidade do modelo decisor, com destaque para o Qwen3-4B, que alcançou mais de 91% em acurácia e completude. Já o gpt-3.5-turbo teve bons resultados pontuais, mas sem consistência geral.

Os resultados também evidenciam um *trade-off* entre qualidade e *overhead*: execuções locais reduzem dependência de rede e são mais adequadas a cenários dinâmicos, enquanto APIs aumentam custos e latência. Como limitações, destacam-se o *overhead* ainda elevado e a sensibilidade a saídas inconsistentes. Como trabalhos futuros, propõem-

se melhorias na validação das decisões, expansão dos testes com maior ambiguidade e estratégias de personalização com aprendizado incremental a partir do *feedback* do usuário.

Disponibilidade de Artefatos

Em linha com princípios e práticas de Ciência Aberta, disponibilizamos a implementação da nossa abordagem através do repositório: `https://github.com/carnotbraun/VAMOS`

Agradecimentos

Este projeto foi apoiado pelo Ministério da Ciência, Tecnologia e Inovações, com recursos da Lei nº 8.248, de 23 de outubro de 1991, no âmbito do PPI-SOFTEX, coordenado pela Softex e publicado Arquitetura Cognitiva (Fase 3), DOU 01245.003479/2024-10. Este trabalho foi parcialmente patrocinado pelo projeto CNPq 407192/2025-5.

Referências

Barbon, R. S., Madeira, E. R., and Akabane, A. T. (2025). A two-context-aware approach for navigation: A case study for vehicular route recommendation. *Ad Hoc Networks*, 166:103655.

Benmessaoud, Y., Cherrat, L., and Ezziyyani, M. (2023). Real-time self-adaptive traffic management system for optimal vehicular navigation in modern cities. *Computers*, 12(4):80.

Braun, C., Jarczewski, R. O., Talasso, G. U., Villas, L. A., and de Souza, A. M. (2025). Beyond shortest path: Agentic vehicular routing with semantic context.

Chen, A., Ge, X., Fu, Z., Xiao, Y., and Chen, J. (2024a). Travelagent: An ai assistant for personalized travel planning. *arXiv preprint arXiv:2409.08069*.

Chen, R., Song, W., Zu, W., Dong, Z., Guo, Z., Sun, F., Tian, Z., and Wang, J. (2024b). An llm-driven framework for multiple-vehicle dispatching and navigation in smart city landscapes. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2147–2153.

de Souza, A. M., Braun, T., Botega, L. C., Cabral, R., Garcia, I. C., and Villas, L. A. (2019). Better safe than sorry: a vehicular traffic re-routing based on traffic conditions and public safety issues. *Journal of Internet Services and Applications*, 10(1):17.

De Souza, A. M., Brennand, C. A., Yokoyama, R. S., Donato, E. A., Madeira, E. R., and Villas, L. A. (2017). Traffic management systems: A classification, review, challenges, and future perspectives. *International Journal of Distributed Sensor Networks*, 13(4):1550147716683612.

de Souza, A. M., Oliveira, H. F., Zhao, Z., Braun, T., Loureiro, A. A., and Villas, L. A. (2022). Enhancing sensing and decision-making of automated driving systems with multi-access edge computing and machine learning. *IEEE Intelligent Transportation Systems Magazine*, 14(1):44–56.

Gong, L., Lin, Y., Zhang, X., Lu, Y., Han, X., Liu, Y., Guo, S., Lin, Y., and Wan, H. (2024). Mobility-llm: Learning visiting intentions and travel preference from human mobility data with large language models. *Advances in Neural Information Processing Systems*, 37:36185–36217.

Huang, Z., Shi, G., and Sukhatme, G. S. (2024). From words to routes: Applying large language models to vehicle routing. *CoRR*, abs/2403.10795.

Irugalbandara, C., Mahendra, A., Daynauth, R., Arachchige, T. K., Dantanarayana, J., Flautner, K., Tang, L., Kang, Y., and Mars, J. (2024). Scaling down to scale up: A cost-benefit analysis of replacing openai's llm with open source slms in production. In *2024 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, pages 280–291.

Jiang, K., Cai, X., Cui, Z., Li, A., Ren, Y., Yu, H., Yang, H., Fu, D., Wen, L., and Cai, P. (2025). Koma: Knowledge-driven multi-agent framework for autonomous driving with large language models. *IEEE Transactions on Intelligent Vehicles*, 10(10):4655–4668.

Luca, M., Barlacchi, G., Lepri, B., and Pappalardo, L. (2021). A survey on deep learning for human mobility.

Ning, Y. and Liu, H. (2024). Urbankgent: A unified large language model agent framework for urban knowledge graph construction. *Advances in Neural Information Processing Systems*, 37:123127–123154.

Paiva, S., Ahad, M. A., Tripathi, G., Feroz, N., and Casalino, G. (2021). Enabling technologies for urban smart mobility: Recent trends, opportunities and challenges. *Sensors*, 21(6):2143.

Qin, Z., Zhang, P., Wang, L., and Ma, Z. (2025). Lingotrip: Spatiotemporal context prompt driven large language model for individual trip prediction. *Journal of Public Transportation*, 27:100117.

Wang, J., Jiang, R., Yang, C., Wu, Z., Onizuka, M., Shibasaki, R., Koshizuka, N., and Xiao, C. (2024). Large language models as urban residents: An llm agent framework for personal mobility generation. In Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., and Zhang, C., editors, *Advances in Neural Information Processing Systems*, volume 37, pages 124547–124574. Curran Associates, Inc.

Wang, R., Zhou, M., Gao, K., Alabdulwahab, A., and Rawa, M. J. (2022). Personalized route planning system based on driver preference. *Sensors*, 22(1).

Wang, X., Fang, M., Zeng, Z., and Cheng, T. (2023). Where would I go next? large language models as human mobility predictors. *CoRR*, abs/2308.15197.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Wu, Z., Peng, R., Han, X., Zheng, S., Zhang, Y., and Xiao, C. (2023). Smart agent-based modeling: On the use of large language models in computer simulations.

Xi, Z., Chen, W., Guo, X., He, W., Ding, Y., Hong, B., Zhang, M., Wang, J., Jin, S., Zhou, E., Zheng, R., Fan, X., Wang, X., Xiong, L., Zhou, Y., Wang, W., Jiang, C., Zou, Y., Liu, X., Yin, Z., Dou, S., Weng, R., Cheng, W., Zhang, Q., Qin, W., Zheng, Y., Qiu, X., Huang, X., and Gui, T. (2023). The rise and potential of large language model based agents: A survey.

Yang, K., Guo, Z., Lin, G., Dong, H., Huang, Z., Wu, Y., Zuo, D., Peng, J., Zhong, Z., WANG, X., Guo, Q., Jia, X., Yan, J., and Lin, D. (2025). Trajectory-LLM: A language-based data generator for trajectory prediction in autonomous driving. In *The Thirteenth International Conference on Learning Representations*.

Zhang, S., Li, J., Shi, L., Ding, M., Nguyen, D. C., Tan, W., Weng, J., and Han, Z. (2024a). Federated learning in intelligent transportation systems: Recent applications and open problems. *IEEE Transactions on Intelligent Transportation Systems*, 25(5):3259–3285.

Zhang, S., Luo, Z., Yang, L., Teng, F., and Li, T. (2024b). A survey of route recommendations: Methods, applications, and opportunities.

Zheng, K., Zheng, Q., Chatzimisios, P., Xiang, W., and Zhou, Y. (2015). Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions. *IEEE Communications Surveys & Tutorials*, 17(4):2377–2396.

Zhou, Y., Lai, S., Han, J., and Liu, H. (2025). An llm-powered cooperative framework for large-scale multi-vehicle navigation.

Zhou, Z., Lin, Y., Jin, D., and Li, Y. (2024). Large language model for participatory urban planning.