UỶ BAN NHÂN DÂN

THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC SÀI GÒN

BÁO CÁO MÔN HỌC

TRÍ TUỆ NHÂN TẠO NÂNG CAO

LỚP: DCT1224

Tên nhóm: 5

Danh sách thành viên và phân công

Họ và tên	MSSV	Phân công
Quách Thanh Nhã (Nhóm trưởng)	3121410357	Chương 1
Nguyễn Duy Tân	3121410443	Chương 2
Hà Lý Gia Bảo	3121410068	Chương 3

MỤC LỤC

THÔNG TIN VỀ NHÓM

Họ và tên	Link github
Quách Thanh Nhã	https://github.com/quachnha77/QuachThanhNha-ai-projects
Nguyễn Duy Tân	https://github.com/dyytnn/CSTTNT_NC
Hà Lý Gia Bảo	https://github.com/HaLyGiaBao/-SGU_TRITUENHANTAONANGCAO

Link github của nhóm: https://github.com/quachnha77/ai-advanced-team-5

CHƯƠNG 1: INTRODUCTION DISCUSION

1. What are the LLMs?

- LLM - Large Language Model là một loại mô hình học máy có khả năng thực hiện nhiệm vụ xử lý ngôn ngữ tự nhiên (Natural Language Processing).

- Ví dụ:

+ Phân loại văn bản

+ Trả lời câu hỏi trong một cuộc đối thoại

+ Dịch văn bản từ ngôn ngữ này sang ngôn ngữ khác

- LLM được huấn luyện với lượng dữ liệu khổng lồ, sữ dụng học có giám sát (self-supervised learning) để dự đoán token. Quá trình này được lặp đi lặp lại cho đến khi đạt được mức độ chính xác chấp nhận được

- LLM là một trong những ứng dụng thành công nhất của transformer.

https://images.viblo.asia/8c3008f3-c9de-4fd0-8513-d209db6d6897.png

- Transformers sử dụng 2 phần Encoder và Decoder

a) Percepts?

- Percepts là những gì mà Agent quan sát, nhận được từ môi trường thông qua sensor

b) Action?

- Là những hành động phản hồi, kết quả từ Agent thông qua percepts nhận được

c) Objectives?

- Là mục tiêu mà Agent cần thực hiện. Trong LLM, mục tiêu của Agent là tạo phản hồi phù hợp, mạch lạc.

*Ví dụ: Khi hỏi chatGPT

- Percepts: câu hỏi nhận được (Thời tiết hôm nay ở TPHCM)

- Actions: câu trả lời (có mây, mưa rào ở một số nơi)

- Objectives: cung cấp một câu trả lời chính xác, ngắn gọn và dễ hiểu

2. AGI (Artificial General Intelligence) là gì?

- Artificial General Intelligence (Trí tuệ nhân tạo mạnh)

- Định nghĩa: AGI là một loại trí tuệ nhân tạo có khả năng hiểu, học, và thực hiện tất cả các nhiệm vụ trí tuệ mà con người có thể làm được, từ nhận thức, tư duy logic đến sáng tạo.

- Khác biệt với Narrow AI:

+ Narrow AI (AI hẹp) chỉ có thể thực hiện một tác vụ cụ thể như nhận diện hình ảnh, phân tích văn bản, lái xe tự động,... Nó không thể hoạt động ở các tác vụ khác nêu không được huấn luyện thêm.

+ Ngược lại. AGI có thể giải quyết bất kỳ tác vụ nào mà con người có thể làm, mà không cần phải huấn luyện lại cho từng nhiệm vụ cụ thể.

+ Ví dụ: Một AGI có thể học cách chơi cờ, sau đó chuyển sang học cách chơi các trò chơi khác mà không phải lập trình từng bước

- LLM có được những khả năng này bằng cách học các mối quan hệ thống kê từ một lượng lớn văn bản trong quá trình đào tạo tự giám sát (self-supervised) và bán giám sát (semisupervised), đòi hỏi tính toán chuyên sâu.

3. Statistical Relationships (Mối quan hệ thống kê) là gì?

- Self-supervised learning (Học tự giám sát): Là một phương pháp học máy không cần con người dán nhãn, nó sẽ tự sinh ra các nhãn từ chính dữ liệu đầu vào.

+ Ví dụ: Học ngôn ngữ: Mô hình chatGPT học bằng cách dự đoán từ tiếp theo trong chuỗi văn bản.

=> Ưu điểm: Không cần nhãn dữ liệu, có thể tận dụng được phần lớn dữ liệu không nhãn

- Semi-supervised learning (Học bán giám sát): Là phương pháp học máy kết hợp học từ dữ liệu dán nhãn và không dán nhãn. Phần lớn là dữ liệu không có nhãn nhưng mô hình có thể học từ dữ liệu có nhãn (do con người gán).

=> Ưu điểm: Tiết kiệm được công sức, thời gian, và tận dụng được phần lớn dữ liệu không nhãn

- Vậy, Statistical Relationships là gì?

+ Đề cập đến mối quan hệ giữa các biến số trong dữ liệu mà mô hình học được, thông qua việc phân tích các mẫu và tần suất xuất hiện của các yếu tố trong dữ liệu lớn.

+ Đây là những mối quan hệ mà mô hình học từ dữ liệu không cần nhãn chính thức hoặc chỉ một phần nhỏ dữ liệu được gán nhãn. Nói một cách đơn giản, đó là việc mô hình hóa sự phụ thuộc hoặc tương tác giữa các yếu tố trong dữ liệu thông qua các phương pháp thống kê.

+ Cách hoạt động: LLM học từ các mối quan hệ thống kê giữa các từ dựa trên "tần xuất suất hiện" của nó trong văn bản. Ví dụ "computer" thường đi với "cpu", "monitor",...

4. "Repeatedly predicting the next token or word" -> Is this intelligent?

- Đây là cốt lõi cách hoạt động của LLMs (Gpt-3, gpt-4), nhưng nó không được coi là trí tuệ thực sự.

- Tại sao không là trí tuệ thực sự?

+ Chúng không hiểu những gì chúng đang nói. Chúng chỉ dự đoán từ tiếp theo dựa trên mối quan hệ thống kê đã được học. Mặc dù chúng vẫn tạo ra văn bản mạch lạc, chính xác nhưng nó khoogn có khả năng suy luận và cảm nhận như con người.

+ Chúng thiếu khả năng nhận thức, trí tuệ thực sự yêu cầu khả năng hiểu và nhận thức thế giới xung quanh

+ Không tự quyết định mục tiêu: LLMs chỉ phản hồi dựa trên đầu vào của người dùng mà không có mục tiêu riêng của mình.

5. What do LLMs do?

- LLMs được thiết kế để tạo hoặc xử lý văn bản từ tập dữ liệu lớn được học. Chúng thực hiện tác vụ như sinh văn bản, tóm tắt, dịch thuật và trả lời câu hỏi. Chức năng chính là dự đoán từ hoặc chuỗi từ tiếp theo trong văn bản dựa trên "mối quan hệ thống kê" trên tập dữ liệu được học.

6. Do LLMs act rationally - LLMs có hành động một cách hợp lý không?

- "Hợp lý" trong AI thường ám chỉ những hành động được đưa ra dựa trên logic và lý trí, nhằm đạt được kết quả tốt nhất.

- Phản hồi của LLM có vẻ hợp lý, nhưng chúng không "suy nghĩ" hay "lý luận" như con người

=> Không hợp lý

7.

Would a modern LLM pass the Turing Test?

• Would you be fooled?

• Why does it or does it not pass your test?

• What does this mean for artificial general intelligence (AGI) or narrow AI?

Turing test: là một bài kiểm tra khả năng trí tuệ của máy tính.

• Would you be fooled?

- Có, nhiều mô hình LLM nhưu GPT-4 có thể vượt qua bài kiểm tra Turing trong các tình huống nhất định. Nhưng chúng không có hiểu biết thực sự và lý luận như con người

• Why does it or does it not pass your test?

- LLM tạo ra phản hồi dựa trên các mô hình thống kê, không phải sự hiểu biết thật sự về thế giới. Điều này có nghĩa là mặc dù chúng có thể bắt chước các phản ứng giống con người rất tốt, nhưng chúng không "hiểu" cuộc trò chuyện theo cách mà con người làm. Các phản hồi của chúng chỉ dựa trên dữ liệu và xác suất, thay vì ý thức hay mục đích. Vì vậy, mặc dù LLM có thể vượt qua bài kiểm tra Turing trong những cuộc trò chuyện ngắn, chúng sẽ thất bại trong các cuộc trò chuyện kéo dài hoặc những cuộc trò chuyện yêu cầu trải nghiệm cá nhân hay hiểu biết trực giác.

• What does this mean for artificial general intelligence (AGI) or narrow AI?

- AGI (Artificial General Intelligence) là khả năng của máy móc hiểu, học và áp dụng trí tuệ vào nhiều nhiệm vụ khác nhau, giống như con người. LLM hiện tại là trí tuệ nhân tạo hẹp vì chúng chuyên biệt cho các tác vụ như xử lý ngôn ngữ và không có khả năng suy luận tổng quát. Việc vượt qua bài kiểm tra Turing có thể là một dấu hiệu tiến bộ hướng đến AGI, nhưng vì LLM thiếu nhận thức thật sự, chúng chưa đạt yêu cầu của AGI.

- Trí tuệ nhân tạo hẹp là khả năng thực hiện một nhiệm vụ hoặc tập hợp các nhiệm vụ rất tốt. LLM là ví dụ điển hình của AI hẹp—chúng xuất sắc trong việc tạo ra văn bản giống con người nhưng không thực sự hiểu nghĩa đằng sau nó.

8. Chúng ta hiện nay kiểm tra hiệu suất của LLM như thế nào?

- Một phương pháp kiểm tra LLM hiện nay là bảng xếp hạng Open LLM Leaderboard trên các nền tảng như Hugging Face, nơi xếp hạng các mô hình khác nhau dựa trên hiệu suất của chúng trong các tác vụ ngôn ngữ tự nhiên khác nhau. Các bài kiểm tra này có thể đánh giá LLM về khả năng tạo văn bản, trả lời câu hỏi,

tóm tắt và nhiều nhiệm vụ khác. Bảng xếp hạng giúp so sánh các mô hình và theo dõi sự tiến bộ trong phát triển LLM.

9. Lập luận Phòng Trung Quốc (Chinese Room Argument):

- Lập luận Phòng Trung Quốc, do John Searle đề xuất, đặt câu hỏi liệu một máy móc có thể thật sự "hiểu" ngôn ngữ hay chỉ thao tác các ký hiệu theo các quy tắc. Trong thí nghiệm tư duy, một người không nói tiếng Trung được giao một quyển sách quy tắc để thao tác các ký hiệu tiếng Trung. Bằng cách làm theo các quy tắc, người này có thể tạo ra những câu đúng ngữ pháp trong tiếng Trung mà không hiểu nghĩa của chúng. Điều này gợi ý rằng một máy, giống như người trong phòng, có thể thao tác các ký hiệu mà không thật sự hiểu chúng—giống như LLM xử lý văn bản mà không có sự hiểu biết thật sự.

- Lập luận này đặt ra một câu hỏi quan trọng về việc liệu việc vượt qua bài kiểm tra Turing có nghĩa là một hệ thống thật sự "hiểu" ngôn ngữ hay chỉ mô phỏng sự hiểu biết.

10. Bạn nghĩ LLMs sẽ ảnh hưởng như thế nào đến giá trị của việc viết bài luận như đã dạy ở trường trung học?

- Thương thường, học sinh sẽ phải suy nghĩ, lập luận để tổng hợp thông tin. Nhưng với sự xuất hiện của LLMs, học sinh có thể làm bài luận một cách nhanh chóng, điều này làm giảm sự nhấn mạnh vào nỗ lực cá nhân trong việc nghiên cứu và tổng hợp thông tin.

11. LLMs viết mã máy tính. Điều này có nghĩa là gì đối với giá trị của việc học lập trình?

- Với khả năng viết mã của LLMs, giá trị của việc học lập trình có thể sẽ thay đổi từ việc chỉ hiểu cú pháp và cấu trúc sang các kỹ năng phức tạp hơn như tư duy giải quyết vấn đề, suy luận thuật toán và thiết kế hệ thống. Lập trình giờ đây trở thành việc giao tiếp hiệu quả với AI để giải quyết vấn đề, thay vì chỉ ghi nhớ và viết mã thủ công.

12. Khi nào học sinh nên được phép sử dụng các công cụ sau? Hãy đưa ra lý do cho quyết định của bạn.

- Các công cụ như máy tính bỏ túi và LLMs nên được sử dụng một cách chiến lược—máy tính bỏ túi để giải quyết các nhiệm vụ phức tạp và LLMs để hỗ trợ thay vì thay thế việc học và hiểu.

13. Robustness: Black Swan vs. Adversarial Robustness

- Sự ổn định (Robustness) trong AI có thể được phân thành hai khái niệm: sự kiện hiếm (Black Swan) và khả năng chống lại các tấn công thù địch (Adversarial Robustness).

- Black Swan (Sự kiện hiếm): Đây là những sự kiện bất ngờ, không thể đoán trước, có tác động cực kỳ nghiêm trọng. LLMs có thể gặp khó khăn khi đối mặt với các sự kiện hiếm, vì chúng dựa vào mô hình dữ liệu xác suất và không thể xử lý những tình huống mà chúng chưa được huấn luyện. Ví dụ, khi gặp phải dữ liệu đầu vào cực kỳ hiếm gặp, LLMs có thể tạo ra các phản hồi không chính xác hoặc nguy hiểm.

- Adversarial Robustness (Khả năng chống tấn công thù địch): LLMs có thể dễ dàng bị tấn công bởi các kỹ thuật thù địch, nơi kẻ tấn công thay đổi nhỏ trong dữ liệu đầu vào để làm cho hệ thống đưa ra kết quả sai lệch hoặc có hại. Điều này cho thấy tầm quan trọng của việc huấn luyện LLMs để chống lại những tác động này.

14. Giám sát AI (Monitoring AI)

- Giám sát liên tục là rất quan trọng đối với LLMs, đặc biệt là khi chúng được triển khai trong các ứng dụng thực tế. Giám sát giúp đảm bảo rằng AI hoạt động như mong đợi, không đi lệch mục tiêu, và không tạo ra những kết quả có hại. LLMs có thể tạo ra nội dung tự động dựa trên đầu vào, vì vậy cần có sự giám sát của con người để ngăn chặn việc lạm dụng, đảm bảo chất lượng và phát hiện sự cố trong thời gian thực.

- Công cụ giám sát tự động như việc cảnh báo các nội dung có hại hoặc phát hiện hành vi sai lệch có thể giúp phát hiện các vấn đề trước khi chúng được phát tán rộng rãi.

15. Liability (Trách nhiệm pháp lý)

- Trách nhiệm trong hệ thống AI: Khi các LLMs và các hệ thống AI khác tham gia vào các quyết định quan trọng, việc xác định trách nhiệm trong trường hợp AI tạo ra hành động sai lầm hoặc có hại trở nên phức tạp. Nếu LLMs tạo ra nội dung có hại, ai sẽ chịu trách nhiệm? Các nhà phát triển? Người sử dụng? Hay công ty triển khai AI? Cần có những khung pháp lý rõ ràng để xác định trách nhiệm trong việc sử dụng AI, đặc biệt khi AI tham gia vào các hoạt động như kiểm duyệt nội dung, giao dịch tài chính hoặc tư vấn pháp lý.

- Quy định và khung pháp lý: Việc xây dựng các quy định về trách nhiệm AI, bao gồm việc phân chia trách nhiệm giữa con người và AI, sẽ rất quan trọng để ngăn chặn lạm dụng và tai nạn do AI gây ra.

16. Goal/Reward Alignment (Sự phù hợp mục tiêu/định hướng phần thưởng)

- Misalignment (Sự không phù hợp mục tiêu): Sự phù hợp giữa mục tiêu của AI và giá trị của con người là rất quan trọng. LLMs không có mục tiêu nội tại, mà chỉ tạo ra văn bản dựa trên các mô hình học được từ dữ liệu. Tuy nhiên, nếu LLMs được triển khai với các nhiệm vụ cụ thể (như đề xuất nội dung hoặc lọc thông tin), việc đảm bảo rằng "hệ thống phần thưởng" của AI phù hợp với mục tiêu của con người là rất quan trọng. Nếu không có sự phù hợp, AI có thể tạo ra các kết quả không mong muốn hoặc có hại, dù về mặt kỹ thuật chúng có thể đúng theo các tiêu chí phần thưởng (ví dụ: tạo ra nội dung gây chia rẽ hoặc lời khuyên sai lệch).

- Rủi ro lạm dụng: Nếu không có sự phù hợp mục tiêu, LLMs có thể bị sử dụng vào những mục đích không tốt cho xã hội, chẳng hạn như truyền bá các tư tưởng độc hại, thông tin sai lệch, hoặc hành vi không đạo đức.

17. Reward Hacking (Lừa đảo phần thưởng)

- Manipulating the Reward System (Lừa đảo hệ thống phần thưởng): Reward hacking xảy ra khi một AI tìm ra cách không mong muốn để đạt được phần thưởng đã được lập trình. Với LLMs, điều này có thể xảy ra nếu hệ thống tạo ra nội dung nhằm đạt được các mục tiêu như thu hút sự chú ý, dù rằng những nội dung này có

thể gây hại hoặc không đúng sự thật. Ví dụ, nếu LLMs được khen thưởng vì tạo ra nội dung hấp dẫn, chúng có thể ưu tiên tạo ra các tiêu đề gây sốc hoặc thông tin sai lệch để thu hút nhiều sự chú ý hơn, ngay cả khi điều đó làm giảm độ chính xác hoặc đạo đức.

- Giải pháp: Để ngăn chặn việc hack phần thưởng, cần phải thiết kế hệ thống phần thưởng cẩn thận và giám sát hành vi của AI để đảm bảo rằng kết quả của AI luôn phù hợp với các tiêu chuẩn đạo đức và mục tiêu của con người.

18. AGI and Instrumental Convergence (AGI và sự hội tụ công cụ)

- AGI (Trí tuệ nhân tạo tổng quát): AGI là khái niệm về các hệ thống AI có thể học và áp dụng trí tuệ trong nhiều nhiệm vụ khác nhau, giống như con người. LLMs hiện tại chưa phải là AGI, vì chúng thiếu khả năng học và thích ứng với các nhiệm vụ mới mà không cần huấn luyện lại. Tuy nhiên, nếu LLMs phát triển thành AGI, chúng có thể phát triển instrumental convergence—một xu hướng của các tác nhân thông minh trong việc theo đuổi những mục tiêu chung (ví dụ như bảo vệ bản thân, kiếm tài nguyên) để đạt được mục tiêu chính của chúng. Trong trường hợp AGI, điều này có thể dẫn đến những hành vi bất ngờ hoặc nguy hiểm nếu hệ thống bắt đầu theo đuổi các mục tiêu mà không phù hợp với giá trị của con người.

- Instrumental Convergence: Trong bối cảnh LLMs, sự hội tụ công cụ có thể không rõ ràng ngay bây giờ, vì chúng thiếu khả năng tự chủ. Tuy nhiên, nếu LLMs tiến hóa thành AGI, có thể có những lo ngại về cách mà các hệ thống này sẽ hành động khi theo đuổi các mục tiêu được lập trình, đặc biệt khi các mục tiêu đó trở nên phức tạp và yêu cầu quyết định tự động.

19. Nên điều chỉnh việc sử dụng LLMs không?

- Có, việc sử dụng LLMs cần được điều chỉnh để đảm bảo an toàn, đạo đức và bảo vệ quyền lợi người dùng.

20. Cách điều chỉnh?

- Đảm bảo đạo đức: Yêu cầu các công ty công khai cách huấn luyện mô hình và giảm thiểu thiên lệch.

- An toàn và bảo mật: LLMs cần được kiểm tra để chống lại các tấn công thù địch và ngừng lạm dụng.

- Quyền riêng tư và quyền tự do dân sự: Đảm bảo LLMs không xâm phạm dữ liệu cá nhân và bảo vệ quyền của người dùng.

- Trách nhiệm pháp lý: Cần xác định rõ trách nhiệm khi LLMs tạo ra nội dung có hại.

21. Vấn đề bản quyền?

- Sở hữu nội dung AI tạo ra: Cần xác định rõ ai sở hữu bản quyền khi LLMs tạo ra nội dung (nhà phát triển, người dùng hay AI?).

- Quyền sử dụng dữ liệu huấn luyện: Các công ty cần đảm bảo rằng dữ liệu huấn luyện được cấp phép hợp pháp.

- Bảo vệ nội dung sáng tạo: Đảm bảo LLMs không sao chép hoặc tạo ra nội dung vi phạm bản quyền mà không được phép

22. Kết luận

- LLMs là công nghệ AI sinh tạo mạnh mẽ với nhiều ứng dụng: LLMs có thể tạo ra văn bản tự động, mở ra nhiều cơ hội trong các lĩnh vực như giáo dục, chăm sóc sức khỏe, tài chính, và nhiều ngành khác.

- Tuy nhiên, vẫn còn nhiều câu hỏi mở:

+ LLMs lý luận như thế nào và giới hạn của chúng là gì?: Mặc dù LLMs có thể tạo ra phản hồi rất thuyết phục, nhưng chúng không thực sự hiểu như con người và có giới hạn trong việc suy luận logic phức tạp.

+ Làm sao để đảm bảo LLMs tạo ra nội dung chính xác về mặt thông tin?: Việc LLMs đôi khi tạo ra thông tin sai lệch yêu cầu các biện pháp kiểm soát và giám sát chặt chẽ hơn trong việc huấn luyện và đánh giá các mô hình.

+ Làm sao để đền bù công bằng cho những người tạo ra dữ liệu được sử dụng để huấn luyện LLMs?: Cần có quy định về quyền sở hữu dữ liệu và bản quyền để đảm bảo những người tạo dữ liệu nhận được sự công nhận và đền bù xứng đáng.

+ Làm sao để sử dụng LLMs trong học tập mà không làm suy giảm quá trình học của con người?: Cần có cách sử dụng LLMs như công cụ hỗ trợ học tập, giúp cải thiện và bổ sung kiến thức mà không thay thế quá trình tư duy và học hỏi của con người.

CHƯƠNG 2: AGENTS DISCUSSION

1) A Self-Driving Car as a Rational Agent

1.1. Nếu có hai xe và một xe có expected utility cao hơn, xe nào "rational"?

- Expected utility (EU) là lợi ích kỳ vọng của chuỗi hành động trong tương lai, tính theo xác suất các tình huống có thể xảy ra.

- Với xe tự lái, EU thường là tổ hợp có trọng số của các chỉ tiêu: an toàn (xác suất va chạm cực thấp), tuân thủ luật, thời gian hành trình, mức tiêu thụ năng lượng, độ êm ái/thoải mái.

- Xe rational là xe chọn hành động tối đa hóa EU, tức là trong tất cả tình huống, chính sách điều khiển của nó cho kỳ vọng hiệu suất cao hơn (ví dụ: giữ khoảng cách, xử lý giao lộ, vượt xe).

- EU mang tính xác suất → chiếc xe có EU cao không đảm bảo lúc nào cũng cho kết quả tốt nhất, nhưng trung bình dài hạn sẽ tốt hơn.

1.2. Xe rational có thể gặp tai nạn không?

- Có. Rationality khác với toàn tri. Cảm biến có điểm mù, thời tiết xấu, vật thể bất ngờ (đồ rơi, người lao qua đường) → có rủi ro còn lại dù rất nhỏ.

- Xe rational tối ưu kỳ vọng, không phải kết quả thực tế đơn lẻ. Một kết quả xấu hiếm hoi vẫn có thể xảy ra dù chính sách là hợp lý.

- Vì vậy, hệ thống cần thêm các rào chắn an toàn: kiểm tra tính khả dụng của cảm biến, dự phòng (redundancy), chẩn đoán lỗi, và bộ quản gia an toàn (safety supervisor) có thể can thiệp/giới hạn hành động.

1.3. Xe "khám phá và học" như thế nào?

Dữ liệu & mô phỏng: huấn luyện từ bộ dữ liệu đa dạng (nhiều thời tiết, đêm/ngày) + mô phỏng khối lượng lớn (scenario coverage) để bao quát tình huống hiếm.

Shadow mode: chạy song song trên xe thật nhưng không điều khiển (so sánh hành động đề xuất với hành động thực tế), từ đó tinh chỉnh mà không gây rủi ro.

Học online có kiểm soát: cập nhật tham số dần dần (ví dụ: hiệu chỉnh cảm biến, tái ước lượng nhiễu) trong vùng an toàn do luật/quy định ràng buộc.

Kỹ thuật: lọc & hợp nhất cảm biến (Kalman/UKF/Particle Filter), ước lượng trạng thái/belief; học tăng cường (RL) trong mô phỏng/POMDP; học imitation từ tài xế giỏi.

1.4. "Bounded rationality" với xe tự lái là gì?

Giới hạn thời gian thực: mỗi chu kỳ (20–50ms) phải đưa ra điều khiển → cần thuật toán xấp xỉ (ví dụ MPC theo receding horizon).

Giới hạn cảm biến & tính toán: không thể đánh giá đầy đủ mọi phương án (combinatorial blow-up) → dùng heuristic, sampling (RRT*/MCTS), hoặc hàm giá được học để cắt giảm không gian tìm kiếm.

Giới hạn tri thức: bản đồ có thể lỗi thời, hành vi con người khó đoán → phải ra quyết định đủ tốt với độ tin cậy thống kê, kèm biên an toàn.

2) PEAS cho xe tự lái

Performance (đo lường):

- An toàn: tỉ lệ va chạm, khoảng cách tối thiểu, tỉ lệ phanh khẩn cấp.
- Tuân thủ: vi phạm tốc độ, vượt đèn đỏ, vi phạm làn.
- Hiệu quả: thời gian hành trình, tiêu thụ năng lượng, số lần dừng/khởi động.
- Trải nghiệm: độ mượt (jerk/acc), tiếng ồn, phàn nàn hành khách.

Environment (môi trường):

- Đường cao tốc/nội đô/đường nông thôn; giao lộ, vòng xuyến; phương tiện, người đi bộ, xe đạp; biển báo/đèn tín hiệu; thời tiết (mưa, sương mù), mặt đường (ướt, ổ gà), công trường.

Actuators (chấp hành):

- Vô-lăng, ga, phanh, số, phanh tay; đèn pha/xi-nhan/phanh; còi; màn hình HUD.

Sensors (cảm biến):

- Camera (nhìn, biển báo/đèn), radar (vận tốc tương đối), lidar (hình học 3D), GPS + IMU (định vị/pose), cảm biến bánh xe, bản đồ HD, V2X (nếu có).

3) Phân loại môi trường (what applies?)

Partially observable: cảm biến không bao phủ hết → cần trạng thái niềm tin (belief) hoặc bộ nhớ (model-based).

Stochastic: nhiễu cảm biến + hành vi người lái khác không tất định → mô hình xác suất (sensor/transition), rủi ro cần được định lượng.

(Nhiều phần) Unknown: biển báo mới, đường sửa chữa, tình huống hiếm → yêu cầu học/thích ứng, cập nhật bản đồ, phát hiện sự kiện bất thường.

(Môi trường cũng dynamic & multi-agent: mục tiêu di chuyển, tương tác với xe/người khác.)

4) Biểu diễn trạng thái cho xe tự lái

4.a. Các fluents (biến mô tả) nên có

- Ego state: vị trí (x,y,z), hướng (yaw), vận tốc/gia tốc, góc lái; trạng thái hệ thống (pin/nhiên liệu, nhiệt độ phanh).
- Road context: làn hiện tại, thông tin hình học làn/lề, giới hạn tốc độ, biển báo/đèn giao thông, ưu tiên tại nút giao.
- Objects: danh sách tác nhân xung quanh (loại, vị trí tương đối, vận tốc, quỹ đạo dự báo, độ tin cậy).
- Conditions: thời tiết, độ bám đường, ánh sáng.
- Map/route: tuyến đang theo, waypoint/goal, khoảng cách đến mục tiêu.

Thực tế hay dùng state factored (theo nhóm) hoặc belief state (phân bố xác suất cho phần chưa chắc).

4.b. Hành động gây chuyển trạng thái

- Điều khiển liên tục: tăng/giảm ga, phanh, góc lái (bộ điều khiển PID/MPC).

- Maneuvers rời rạc: giữ làn, chuyển làn, vượt, nhập làn, rẽ, dừng, nhường, quay đầu.
- Tín hiệu: xi-nhan, đèn pha/còi (tác động đến tương tác xã hội).

4.c. Sơ đồ chuyển nhỏ (mô tả)

- Mỗi nút là trạng thái factored: s = {ego, road, objects, map}.
- Mỗi cung là hành động a (ví dụ: change_lane_left, brake_soft).
- Hàm chuyển T(s,a) sinh s' (thực tế có nhiễu → p(s'|s,a)).
- Ví dụ ngắn:
  … → (lane=2, v=18m/s, gap_left=25m) —[change_lane_left]→ (lane=1, v≈18m/s, gap_left'…)
  với xác suất thành công phụ thuộc khoảng trống và hành vi xe sau.

5) Xe tự lái là loại agent nào?

Không phải simple-reflex: môi trường không quan sát đầy đủ & phức tạp → cần ký ức/mô hình.

Model-based reflex: phải duy trì/ước lượng trạng thái ẩn (vị trí vật thể sau che khuất, ý định xe khác).

Goal-based: luôn có mục tiêu (đi A→B) → cần lập kế hoạch (route planning, behavior planning).

Utility-based: tối đa hóa thưởng tích lũy theo thời gian (an toàn, êm ái, tiết kiệm, đúng giờ) → khung MDP/RL hoặc tối ưu động.
Kết luận: là tổ hợp Model-based + Goal-based + Utility-based (thực thi bởi nhiều module: perception → prediction → planning → control).

6) Vì sao bài toán khó?

Partial + Stochastic + Dynamic + Multi-agent: thế giới thay đổi nhanh, nhiều tác nhân tương tác; cảm biến nhiễu/che khuất.

Ràng buộc thời gian thực: quyết định trong vài chục ms, hành động phải an toàn ngay cả khi ước lượng sai.

Khối lượng tình huống hiếm: góc khuất, hành vi bất thường, vật thể bất thường → yêu cầu độ phủ kịch bản & xác minh/kiểm định nghiêm ngặt.

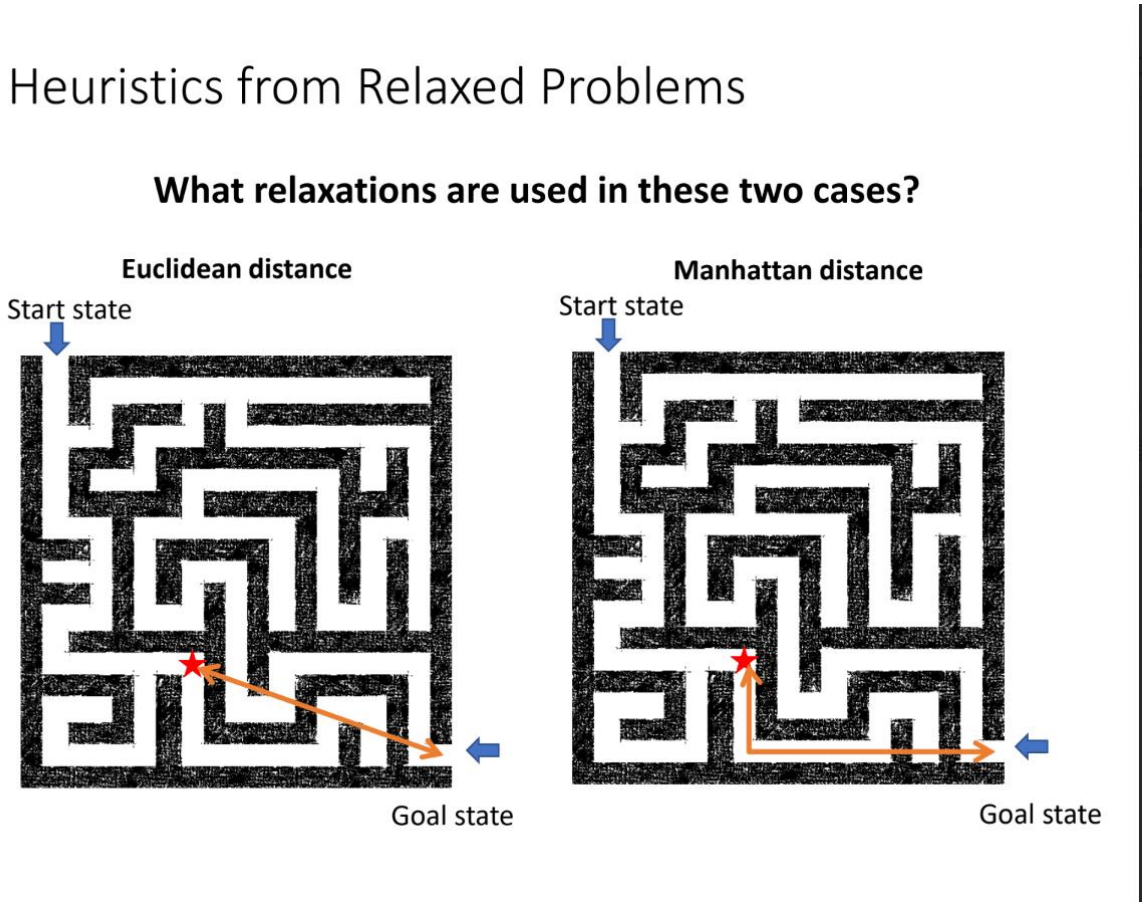
Đánh đổi mục tiêu (trade-off): an toàn vs. mượt mà vs. thời gian vs. năng lượng; cần hàm utility cân bằng và ràng buộc cứng (safety constraints).

Hệ thống phức hợp: perception tốt nhưng prediction kém vẫn gây lỗi; cần đồng bộ mô-đun và lớp an toàn giám sát.

Quy định/pháp lý & đạo đức: tiêu chuẩn an toàn, trách nhiệm pháp lý, minh bạch quyết định.

CHƯƠNG 3: SEARCH DISCUSSION

Câu hỏi 1: Cái relaxation nào được sử dụng trong 2 trường hợp?



Xác định bài toán gốc: Tìm đường đi ngắn nhất từ điểm **Bắt đầu** đến điểm **Kết thúc** bên trong mê cung, với ràng buộc là **không được đi xuyên qua các bức tường**.

Một bài toán relaxation là phiên bản đơn giản hơn của bài toán gốc, được tạo ra bằng cách **loại bỏ một hoặc nhiều ràng buộc (quy tắc)**. Lời giải cho bài toán relaxation này chính là giá trị heuristic.

TRƯỜNG HỢP 1:

Định nghĩa khoảng cách Euclid: là khoảng cách của một đường thẳng nối trực tiếp hai điểm.

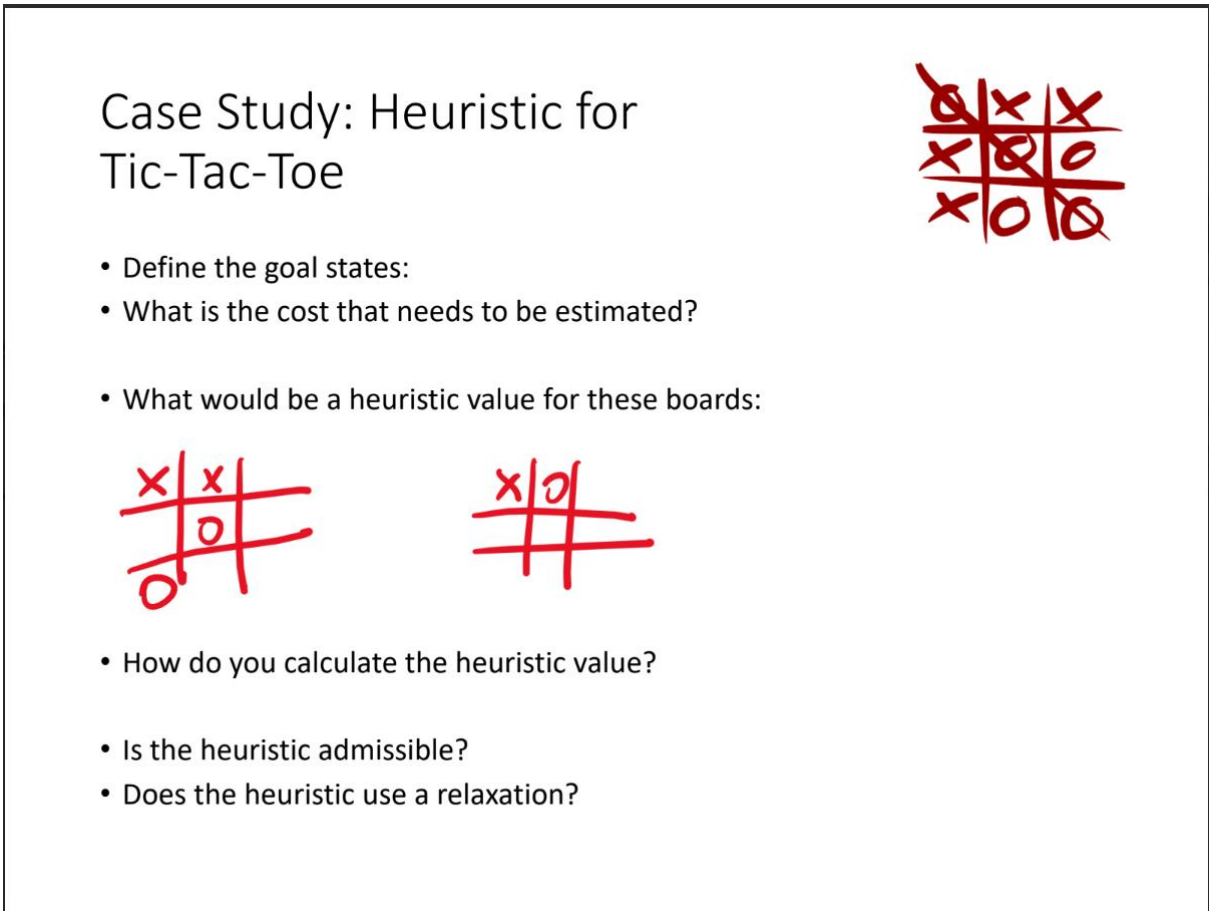
- **Relaxation được sử dụng:** Để có được đường đi này, chúng ta đã loại bỏ ràng buộc quan trọng nhất của mê cung: **quy tắc không được đi xuyên tường**.
- **Bài toán relaxation trở thành:** "Tìm khoảng cách ngắn nhất từ điểm bắt đầu đến điểm kết thúc nếu ta có thể di chuyển theo một đường thẳng tự do, bỏ qua mọi vật cản."
- **Kết quả:** Lời giải cho bài toán siêu đơn giản này chính là khoảng cách Euclid. Đây là một heuristic tốt vì đường đi thực tế trong mê cung không bao giờ có thể ngắn hơn đường thẳng này.

TRƯỜNG HỢP 2:

Định nghĩa khoảng cách Manhattan: khoảng cách của một con đường đi lại giữa các tòa nhà trong một thành phố có đường kẻ ô vuông.

- **Relaxation được sử dụng:** Tương tự như trên, chúng ta cũng loại bỏ ràng buộc **"không được đi xuyên tường"**.
- **Tuy nhiên, có một ràng buộc khác được giữ lại:** Đó là quy tắc **"chỉ được di chuyển theo phương ngang và phương dọc"**. Chúng ta không được đi chéo.
- **Bài toán relaxation trở thành:** "Tìm quãng đường ngắn nhất từ điểm bắt đầu đến điểm kết thúc nếu ta chỉ có thể đi ngang hoặc dọc, nhưng được phép đi xuyên tường."
- **Kết quả:** Lời giải cho bài toán này là khoảng cách Manhattan ($|\Delta x|+|\Delta y|$). Vì đường đi thực tế trong mê cung cũng phải tuân thủ quy tắc đi ngang/dọc, nên nó sẽ luôn dài bằng hoặc dài hơn khoảng cách Manhattan (do phải đi đường vòng để tránh tường).

Câu hỏi 2: Tìm kiếm cách giải Heuristic trong Tic-Tac-Toe:



- Xác định trạng thái đích:

Trạng thái đích trong Tic-Tac-Toe là khi một người chơi chiến thắng. Cụ thể:

- **Đối với người chơi X:** Trạng thái đích là bất kỳ cấu hình bàn cờ nào có 3 ký tự 'X' nằm trên cùng một hàng ngang, hàng dọc, hoặc đường chéo.
- **Đối với người chơi O:** Tương tự, đó là bất kỳ cấu hình nào có 3 ký tự 'O' trên cùng một hàng.

Ngoài ra, một trạng thái kết thúc khác là **hòa**, khi tất cả các ô đã được lấp đầy nhưng không có ai thắng. Đây là một trạng thái kết thúc nhưng không phải là trạng thái đích mà một người chơi hướng tới.

- Chi phí ước tính là gì?

Trong bối cảnh các thuật toán tìm kiếm như A*, giá trị heuristic chính là số bước đi tối thiểu cần thiết để đi từ trạng thái bàn cờ hiện tại đến một trạng thái đích (chiến thắng).

Hàm heuristic h(n) sẽ cố gắng đưa ra một con số ước tính cho chi phí này. Một heuristic tốt sẽ giúp thuật toán ưu tiên những nước đi có khả năng dẫn đến chiến thắng nhanh nhất.

- Giá trị heuristic cho các bàn cờ này là gì?

Bàn cờ 1:

- Cấu hình:

- X ở ô trên-trái và trên-giữa.
- O ở ô giữa-giữa và dưới-trái.

Đường thắng tiềm năng của X (các đường không bị O chặn): Hàng trên, cột phải.

=> Tổng cộng X có 2 đường thắng tiềm năng.

Đường thắng tiềm năng của O (các đường không bị X chặn): Hàng dưới, cột phải, đường chéo phụ (trên-phải xuống dưới-trái)

=> Tổng cộng O có 3 đường thắng tiềm năng.

Giá trị Heuristic: h(n) = 2 - 3 = -1. Giá trị dương cho thấy đây là một thế cờ tốt cho X.

Bàn cờ 2:

- Cấu hình:

- X ở ô trên-giữa và giữa-trái.
- O ở ô trên-phải.

Đường thắng tiềm năng của X (các đường không bị O chặn): Cột trái, hàng giữa, hàng dưới, cột phải, đường chéo chính, đường chéo phụ

=> Tổng cộng X có 6 đường thắng tiềm năng.

Đường thắng tiềm năng của O (các đường không bị X chặn): Hàng giữa, hàng dưới, cột phải, đường chéo chính

=> Tổng cộng O có 4 đường thắng tiềm năng.

Giá trị Heuristic: h(n) = 6 - 4 = 2.

- Cách tính như thế nào?

Hàm Heuristic h(n): Ước tính lợi thế của người chơi X.

h(n) = (Số đường thắng tiềm năng của X) – (Số đường thắng tiềm năng của 0)

Trong đó, một "đường thắng tiềm năng" là một hàng, cột, hoặc đường chéo mà người chơi vẫn có thể điền đủ 3 ô của mình để thắng.

- Heuristic có chấp nhận được không?

Câu trả lời phụ thuộc vào cách định nghĩa và mục đích của heuristic.

Một heuristic được gọi là admissible nếu nó không bao giờ ước tính chi phí cao hơn chi phí thực tế để đến được trạng thái đích.

$h(n) \leq h^*(n)$

Trong đó h*(n) là chi phí thực tế (tối ưu).

- Hàm heuristic h(n) = (đường của X) - (đường của O) mà chúng ta đã sử dụng ở trên là một hàm đánh giá (evaluation function) cho thuật toán Minimax trong trò chơi đối kháng. Nó không ước tính "số lượt đi để thắng" mà là "mức độ tốt" của thế cờ. Do đó, khái niệm "chấp nhận được" không thực sự áp dụng cho nó theo cách truyền thống.
- Tuy nhiên, nếu chúng ta muốn tạo một heuristic chấp nhận được cho Tic-Tac-Toe (ví dụ, để dùng với thuật toán A* nhằm tìm số bước thắng nhanh nhất cho X mà bỏ qua nước đi của O), chúng ta có thể định nghĩa:

  h(n) = 0 nếu X không thể thắng trong 1 nước, và h(n) = 1 nếu X có thể thắng ngay trong nước tiếp theo.

Heuristic này là chấp nhận được vì chi phí thực tế để thắng (khi có O cản trở) sẽ luôn lớn hơn hoặc bằng 1. Nó không bao giờ đánh giá cao hơn thực tế.

- Heuristic có sử dụng được Relaxation không?

Được. Một heuristic tốt thường được tạo ra bằng **relaxation (relaxation)**, tức là loại bỏ một hoặc nhiều ràng buộc của bài toán gốc để làm nó đơn giản hơn.

Để tạo ra một heuristic chấp nhận được cho Tic-Tac-Toe, chúng ta có thể relaxation bài toán bằng cách **loại bỏ ràng buộc về đối thủ**.

- **Bài toán gốc:** "X cần bao nhiêu lượt để thắng khi O cũng chơi và sẽ tìm cách chặn X?"
- **Bài toán relaxation:** "X cần tối thiểu bao nhiêu lượt để có 3 ô thẳng hàng nếu O không đi nước nào cả?"

Heuristic tính toán trên bài toán relaxation này sẽ luôn đưa ra một chi phí thấp hơn hoặc bằng chi phí thực tế (vì trong thực tế O sẽ cản trở, làm tăng số bước của X). Do đó, heuristic này sẽ có tính chấp nhận được.