Con rắn Mamba trong loài Llama: Chưng cất và tăng tốc các mô hình lai

Junxiong Wang*1 , Daniele Paliotta 2,3, Avner May3 , Alexander M. Rush1 ,và Trí Đạo3,4

1Đại học Cornell
2Đại học Geneva
3Cùng nhau AI
4Đại học Princeton

Tóm tắt

Kiến trúc RNN tuyến tính, như Mamba, có thể cạnh tranh với các mô hình Transformer trong mô hình ngôn ngữ trong khi vẫn có các đặc điểm triển khai có lợi. Với trọng tâm là đào tạo các mô hình Transformer quy mô lớn, chúng tôi xem xét thách thức trong việc chuyển đổi các mô hình được đào tạo trước này để triển khai. Chúng tôi chứng minh rằng có thể chắt lọc các Transformer lớn thành RNN tuyến tính bằng cách sử dụng lại các trọng số chiếu tuyến tính từ các lớp chú ý với các tài nguyên GPU học thuật. Mô hình lai kết quả, kết hợp một phần tư các lớp chú ý, đạt được hiệu suất tương đương với Transformer gốc trong các điểm chuẩn trò chuyện và vượt trội hơn các mô hình Mamba lai nguồn mở được đào tạo từ đầu với hàng nghìn tỷ mã thông báo trong cả điểm chuẩn trò chuyện và điểm chuẩn chung. Hơn nữa, chúng tôi giới thiệu một thuật toán giải mã suy đoán nhận biết phần cứng giúp tăng tốc độ suy luận của Mamba và các mô hình lai. Nhìn chung , chúng tôi chỉ ra cách chúng tôi có thể loại bỏ nhiều lớp chú ý ban đầu và tạo ra từ mô hình kết quả hiệu quả hơn với các tài nguyên tính toán hạn chế. Mô hình hiệu suất cao nhất của chúng tôi, được chắt lọc từ Llama3-8B-Instruct, đạt tỷ lệ thắng kiểm soát độ dài 29,61 trên AlpacaEval 2 so với GPT-4 và 7,35 trên MT-Bench, vượt qua mô hình RNN tuyến tính được điều chỉnh theo hướng dẫn tỷ lệ 8B tốt nhất. Chúng tôi cũng thấy rằng mô hình chắt lọc có

ngoại suy chiều dài, cho thấy độ chính xác gần như hoàn hảo trong thử nghiệm kim trong đống cỏ khô ở độ dài chưng cất gấp 20 lần. Mã và các điểm kiểm tra được đào tạo trước là nguồn mở tại https://github.com/jxiw/MambaInLlama và https://github.com/itsdaniele/speculative_mamba.

1 Giới thiệu

Trong khi Transformers [73] là một kiến trúc thiết yếu trong học sâu và đã thúc đẩy sự thành công của các mô hình ngôn ngữ lớn như GPT [9], Llama [71] và Mistral [37], chúng lại quá chậm đối với việc tạo chuỗi rất dài do độ phức tạp bậc hai của chúng liên quan đến độ dài chuỗi và yêu cầu bộ đệm khóa-giá trị (KV) lớn. Các mô hình RNN tuyến tính gần đây (Mamba [26], Mamba2 [18], GLA [79], RWKV [55], RetNet [68], Griffin [19]) đã đánh bại Transformers trong các thí nghiệm được kiểm soát ở quy mô nhỏ đến trung bình, mặc dù các Transformers tốt nhất vẫn vượt trội hơn đáng kể so với các mô hình này trong các tác vụ hạ lưu. Chúng tôi lưu ý rằng thời gian đào tạo của các mô hình RNN tuyến tính tương tự như thời gian đào tạo của các Transformers được tối ưu hóa cao [79], do đó việc mở rộng quy mô bất kỳ mô hình nào trong số này đều yêu cầu tài nguyên tính toán đáng kể.

Lợi ích chính của các mô hình RNN tuyến tính (Mamba [26], Mamba2 [18]) là chúng có suy luận nhanh hơn (thông lượng cao hơn 5 lần) so với Transformers. Suy luận hiệu quả đang nổi lên như một nhu cầu quan trọng đối với các hệ thống LLM như các ứng dụng mới hiện đang bị tắc nghẽn bởi bộ đệm KV lớn của Transformers, ví dụ như suy luận trên nhiều tài liệu dài [30, 56, 65] và các tệp trong cơ sở dữ liệu mã lớn [42, 61]). Các quy trình làm việc mới nổi với các tác nhân [77, 81] cũng yêu cầu suy luận theo lô lớn để khám phá nhiều quỹ đạo hơn và ngữ cảnh dài để mô hình hóa các môi trường phức tạp.

*Đóng góp bằng nhau. Thứ tự được xác định bằng cách tung đồng xu.

Các đặc tính này thúc đẩy mục tiêu chưng cất một mô hình Transformer được đào tạo trước lớn thành một RNN tuyến tính để tạo ra hiệu quả nhất có thể. Có hai thách thức về mặt kỹ thuật: cách ánh xạ các trọng số Transformer được đào tạo trước thành các trọng số RNN tuyến tính để chưng cất và cách điều chỉnh các kỹ thuật suy luận Transformer thực hành tốt nhất, chẳng hạn như giải mã suy đoán, cho kiến trúc mới. Chúng tôi đưa ra những đóng góp sau: •

Chúng tôi chỉ ra rằng bằng cách sử dụng lại các trọng số từ các lớp chú ý, có thể chưng cất một transformer lớn thành một RNN lai-tuyến tính lớn với lượng tính toán bổ sung tối thiểu trong khi vẫn giữ nguyên phần lớn chất lượng tạo ra của nó. Chúng tôi đề xuất một kiến trúc Mamba đã sửa đổi có thể được khởi tạo trực tiếp từ khối chú ý của một mô hình được đào tạo trước.

• Chúng tôi đề xuất một phương pháp chưng cất nhiều giai đoạn phản ánh đường ống LLM tiêu chuẩn kết hợp chưng cất lũy tiến, tinh chỉnh có giám sát [39] và tối ưu hóa sở thích có hướng [58]. Phương pháp này cho thấy sự phức tạp và đánh giá hạ nguồn tốt hơn so với chưng cất vani. • Chúng tôi phát triển một thuật toán lấy mẫu suy đoán nhận biết phần cứng và một hạt nhân nhanh để giải mã suy đoán

trên kiến trúc Mamba và lai. Chúng tôi đạt được thông lượng hơn 300 mã thông báo/giây cho mô hình Mamba 7B. Ngoài ra, chúng tôi chỉ ra rằng giải mã suy đoán có thể được áp dụng hiệu quả cho kiến trúc lai của chúng tôi.

Các thí nghiệm của chúng tôi chưng cất các LLM trò chuyện mở quy mô lớn khác nhau, Zephyr-7B [72], Llama-3 8B [21] thành các mô hình RNN tuyến tính (Mamba lai và Mamba2), chỉ sử dụng 20B mã thông báo đào tạo. Kết quả cho thấy phương pháp chưng cất phù hợp với mô hình giáo viên trong các điểm chuẩn Chat tiêu chuẩn [43, 84]. Chúng tôi cũng chỉ ra rằng nó hoạt động ngang bằng hoặc tốt hơn với tất cả các mô hình Mamba được đào tạo trước có kích thước tương tự từ đầu bao gồm các mô hình Mamba 7B [26, 52] được đào tạo từ đầu với 1,2T mã thông báo hoặc các mô hình NVIDIA Hybrid Mamba2 [74] được đào tạo từ đầu với 3,5T mã thông báo trong nhiều tác vụ (ví dụ: MMLU [34], TruthfulQA [47]) trong đánh giá LM [25].
Đồng thời với công trình này, MOHAWK [6] đã chắt lọc một biến thể Mamba-2 dựa trên kiến trúc Phi-1.5 với chi phí tính toán và giảm hiệu suất hạn chế.

2 Từ Transformer đến Mamba

2.1 Mối quan hệ giữa sự chú ý và RNN tuyến tính

Chúng tôi bắt đầu bằng cách xem xét sự chú ý của nhiều đầu để làm rõ hình dạng của các đối tượng trung gian. Về mặt ký hiệu, chúng tôi sử dụng các chỉ số dưới rõ ràng cho vị trí chuỗi thay vì biểu diễn ma trận, để làm nổi bật hơn những điểm tương đồng giữa hai mô hình.

Sự chú ý được tính toán song song cho nhiều đầu được tham số hóa khác nhau. Mỗi đầu lấy chuỗi o với kích thước ẩn D làm đối số và tính toán,

$$Q_t = W_Q o_t,\ K_t = W_K o_t,\ V_t = W_V o_t \quad \text{cho tất cả } t,$$

$$\alpha_{1} \ldots \alpha_{T} = \text{softmax}\left[m_{1,t} q_t^\top K_1 \ldots m_{T,t} q_t^\top K_T\right] / \sqrt{D} \qquad y_t = \sum_{s=1}^{t} \alpha_s V_s$$

trong đó $o_t \in \mathbb{R}^{D\times 1}$, $W_* \in \mathbb{R}^{D\times D}$, $q_t, K_t, V_t \in \mathbb{R}^{\text{Số}\times 1}$, $m_{s,t} = 1(s \le t)$

Các công trình nghiên cứu gần đây cho rằng RNN tuyến tính có thể là đối thủ cạnh tranh nghiêm trọng đối với sự chú ý trong các mô hình ngôn ngữ lớn. Một số công thức RNN tuyến tính khác nhau đã được đề xuất với các công thức tương tự. Hiện tại, chúng tôi để lại các hình dạng của các tham số A_t, B_t, C_t trừu tượng và lưu ý rằng tất cả các RNN tuyến tính đều có dạng sau đây, ánh xạ một chuỗi 1 chiều sang một chuỗi khác thông qua trạng thái ẩn h có giá trị ma trận ngầm định.

$$h_t = A_t h_{t-1} + B_t x_t, \quad y_t = C_t^\top h_t \qquad (1)$$

RNN tuyến tính có một số lợi thế về mặt tính toán so với attention. Trong quá trình đào tạo, tất cả các giá trị y_t có thể được tính toán hiệu quả hơn attention vì không có softmax phi tuyến tính. Trong quá trình suy luận, mỗi y_t tiếp theo có thể được tính toán tuần tự mà không cần bộ nhớ đệm.

Mặc dù có hình thức khác biệt nhưng vẫn có mối quan hệ tự nhiên giữa RNN tuyến tính và sự chú ý. Tuyến tính hóa công thức chú ý bằng cách loại bỏ softmax mang lại:

$$yt = \sum_{s=1}^{t} \alpha svs = \frac{1}{\sqrt{D}} \sum_{s=1}^{t} ms,tQ \; t \; Ksvs = \frac{1}{\sqrt{D}} \sum_{s=1}^{t} c\hat{o},tKsvs$$

Điều này ngụ ý rằng tồn tại một dạng RNN tuyến tính của sự chú ý tuyến tính, cụ thể:

$$ht = mt \; 1,tht \; 1 + Ktvt \; yt = ht \sqrt{D} \; \frac{1}{\sqrt{D}} Q \; t$$

$$ht = Atht \; 1 + Btxt, \; yt = C \; ht \; t$$

$$At = mt \; 1,t, \; Bt = WKot, \; Ct = WQot, \; xt = WV \; ot$$

Tuy nhiên, lưu ý rằng phiên bản này sử dụng trạng thái ẩn có kích thước h $R^{Số \times 1}$. Chỉ theo dõi hiệu quả một số vô hướng trên thời gian trên mỗi chiều ẩn. Á p dụng phép biến đổi này một cách ngây thơ sẽ dẫn đến kết quả kém. Vấn đề là việc tuyến tính hóa sự chú ý tạo ra biểu diễn bị suy giảm của mô hình ban đầu, vì tính phi tuyến tính softmax rất quan trọng đối với sự chú ý.

Chìa khóa để cải thiện các mô hình này là tăng khả năng của trạng thái ẩn tuyến tính để nắm bắt tốt hơn cấu trúc dài hạn. Ví dụ, công trình trước đây đã chỉ ra việc sử dụng các phương pháp hạt nhân để cải thiện phép xấp xỉ này [36, 63, 83]. Các cách tiếp cận này mở rộng kích thước của biểu diễn trạng thái ẩn thành h đến N×N′ R

để phù hợp hơn với khả năng mô hình hóa của softmax.

2.2 Chưng cất thành RNN tuyến tính mở rộng

Để thiết kế một RNN tuyến tính chưng cất hiệu quả, chúng tôi hướng đến mục tiêu bám sát nhất có thể vào tham số hóa Transformer ban đầu, đồng thời mở rộng khả năng của RNN tuyến tính theo cách hiệu quả. Chúng tôi sẽ không cố gắng để mô hình mới nắm bắt chính xác hàm chú ý ban đầu, mà thay vào đó sử dụng dạng tuyến tính làm điểm khởi đầu cho quá trình chưng cất.

Cụ thể, chúng tôi điều chỉnh tham số hóa từ Mamba, [26] để tăng kích thước trạng thái ẩn, trong khi khởi tạo từ biểu diễn sự chú ý.

Mamba sử dụng mô hình không gian trạng thái thời gian liên tục (SSM) để tham số hóa RNN tuyến tính tại thời điểm chạy, được mô tả bằng phương trình vi phân,

$$h'(k) = Ah(k) + B(k)x(k) \; y(k) = C(k)h(k)$$

Trong đó A là ma trận đường chéo và các giá trị khác là tín hiệu liên tục. Để áp dụng công thức này vào bài toán thời gian rời rạc như mô hình ngôn ngữ, chúng tôi sử dụng mạng nơ-ron để tạo ra chuỗi các khoảng thời gian lấy mẫu t và các mẫu tín hiệu tại các bước thời gian này. Với các khoảng thời gian lấy mẫu này và T mẫu của B, C, Mamba xấp xỉ phương trình thời gian liên tục bằng cách sử dụng RNN tuyến tính làm phép rời rạc. Chúng tôi sử dụng thanh ngang để chỉ dạng thời gian rời rạc, được tái tạo một cách động .

Thuật toán 1 Mamba khởi tạo sự chú ý

```
Hình dạng: B - Lô, L - Chiều dài, D - kích thước nhúng,
          N = D/Đầu, N′ - mở rộng
Đầu vào: ot: (B, D)
Đầu ra: đầu ra: (B, D)
Tham số mới: MLP, A
đối với mỗi đầu Wk , Wq , Wv , Wo : (N, D) mở
          rộng các nhóm KV làm
Tham số đầu: A : (N, N′ )

        cho tất cả các vị
        trí t: xt : (B, N)   WV ot
        Bt : (B, N)   WKot
        Ct : (B, N)   WQot  t :
        (B, N′ )   MLP(xt)
        A1: T′B1: T′C1:T : (B, N, N′ )   Đĩa(A, B, C,  )
y   LinearRNN(A, B, C, x)
đầu ra   đầu ra + WO  y
trả về đầu ra
```

$$\overline{A1...T}, \overline{B1...T}, \overline{C1...T} = Rời rạc(A, B1...T, C1...T, 1...T)$$

Trong trường hợp đơn giản nhất này, với N′ = 1 và một phép rời rạc danh tính, cách tiếp cận này khôi phục sự chú ý tuyến tính đến chuyển đổi RNN tuyến tính đã thảo luận trong phần trước. Lợi ích của Mamba là với N′ > 1, tham số hóa thời gian liên tục cho phép mô hình học các hàm phong phú hơn đáng kể, mà không cần nhiều tham số hơn hoặc giảm hiệu quả. Cụ thể, các tham số học được bổ sung duy nhất sẽ là

tỷ lệ lấy mẫu và A động. Các tham số mới này sẽ kiểm soát RNN tuyến tính được xây dựng thông qua hàm rời rạc tạo ra RNN tuyến tính có giá trị ma trận mới. Cụ thể, chúng tôi lấy cùng một giá trị, làm tăng hệ số của N' $N\times1$ và t R nhưng đầu ra $\bar{B}t, \bar{C}t$ R $\bar{B}t, \bar{C}t$ $R^{N'\times N\times1}$ hiệu quả kích thước ẩn lên trên sự chú ý tuyến tính ngây thơ.

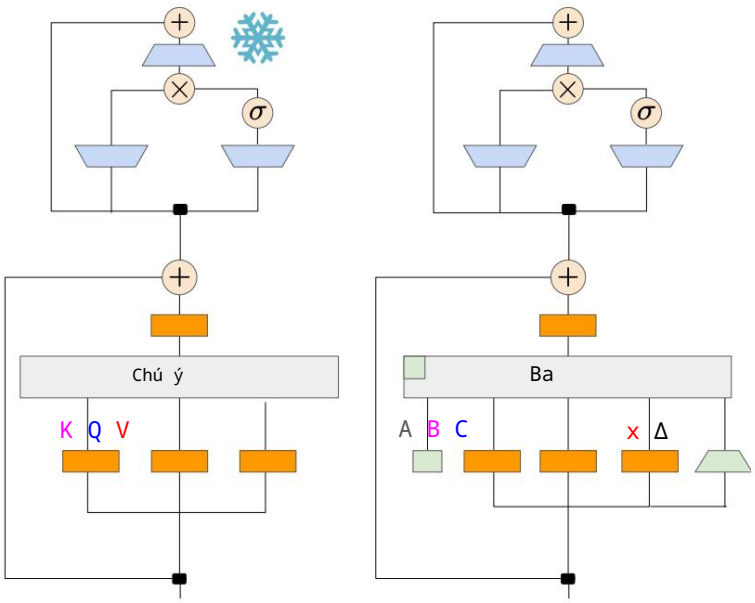
Một đóng góp cốt lõi của Mamba [18, 26] là chứng minh khả năng phân tích phần cứng của thuật toán này. Việc triển khai thuật toán một cách ngây thơ sẽ chậm một cách vô lý vì các tham số mở rộng mới khá lớn. Cách tiếp cận của họ kết hợp sự rời rạc, mở rộng trạng thái và áp dụng RNN tuyến tính vào một hạt nhân duy nhất, điều này tránh được việc hiện thực hóa hoàn toàn các tham số rời rạc. Điều này cho phép N' lớn với chi phí hiệu quả tương đối nhỏ.

2.3 Khởi tạo Attention-to-Mamba và huấn luyện từng bước lai Cách tiếp cận đầy đủ của

chúng tôi được thể hiện trong Thuật toán 1. Thuật toán này đưa các đầu Q, K, V chuẩn từ attention trực tiếp vào phép rời rạc Mamba, sau đó áp dụng RNN tuyến tính kết quả. Như đã lưu ý ở trên, điều này có thể được coi là khởi tạo sơ bộ với attention tuyến tính và cho phép mô hình học các tương tác phong phú hơn thông qua trạng thái ẩn mở rộng.

Hình 1 cho thấy kiến trúc kết quả. Phiên bản của chúng tôi thay thế trực tiếp các đầu chú ý Transformer bằng các lớp RNN tuyến tính tinh chỉnh. Chúng tôi giữ nguyên các lớp MLP Transformer và không đào tạo chúng. Cách tiếp cận này cũng yêu cầu xử lý các thành phần bổ sung như sự chú ý truy vấn nhóm chia sẻ khóa và giá trị giữa các đầu. Chúng tôi lưu ý rằng kiến trúc này khác với kiến trúc được sử dụng trong nhiều hệ thống Mamba , kết hợp các lớp MLP-SSM và sử dụng một đầu duy nhất.

Khởi tạo này cho phép chúng ta thay thế bất kỳ khối chú ý nào bằng khối RNN tuyến tính. Chúng tôi thử nghiệm với các mô hình lai trong đó chúng tôi giữ lại mọi lớp chú ý n . Theo kinh nghiệm, chúng tôi thấy rằng thay thế các lớp theo cách từng bước là chiến lược hiệu quả nhất, tức là trước tiên chúng tôi giữ lại mọi lớp 2, chưng cất, sau đó là mọi lớp 4, và tiếp tục chưng cất.



Hình 1: Chuyển Transformer sang Mamba. Trọng số, màu cam, được khởi tạo từ Transformer (Các phép chiếu tuyến tính cho Q, K và V được khởi tạo bằng phép chiếu tuyến tính cho C, B và X tương ứng). Chúng tôi thay thế các đầu chú ý riêng lẻ bằng các đầu Mamba, sau đó tinh chỉnh các khối Mamba trong khi đóng băng các khối MLP. Các hình dạng chủ yếu được giữ nguyên. Các trọng số màu xanh lá cây được thêm vào. Các tham số mới được giới thiệu cho các tham số A và đã học.

3 Chắt lọc kiến thức cho LM liên kết

Chưng cất kiến thức (KD) [35] đóng vai trò là một kỹ thuật nén nhằm mục đích đào tạo một mạng nhỏ hơn mô phỏng hành vi của một mạng giáo viên lớn hơn. Sau khi khởi tạo mô hình từ các tham số Transformer, chúng tôi hướng đến việc chưng cất nó để thực hiện ngang bằng với mô hình ngôn ngữ gốc. Chúng tôi giả định rằng hầu hết kiến thức từ transformer được duy trì trong các lớp MLP được chuyển từ mô hình gốc và tập trung vào việc chưng cất các bước tinh chỉnh và căn chỉnh của LLM. Trong giai đoạn này, các lớp MLP được giữ nguyên và các lớp Mamba được đào tạo như trong Hình 1.

Điều chỉnh tinh chỉnh có giám sát Đầu tiên, chúng tôi áp dụng chưng cất kiến thức để làm lại giai đoạn điều chỉnh tinh chỉnh có giám sát (SFT) của quá trình điều chỉnh mô hình ngôn ngữ. Trong giai đoạn này, một LLM được đào tạo để tối đa hóa khả năng phản hồi y khi đưa ra lời nhắc nhập x, tức là p(y | x). Nhiệm vụ này trông giống như tạo điều kiện.

Có hai cách tiếp cận phổ biến để chưng cất trong bối cảnh này. Một phương pháp là sử dụng KL-Divergence ở cấp độ từ. Trong bối cảnh này, phân phối xác suất đầy đủ của mô hình học sinh p(·; θ) được đào tạo để khớp với phân phối đầy đủ của mô hình giáo viên p(·; θT) bằng cách giảm thiểu sự phân kỳ KL trên toàn bộ tập hợp các mã thông báo có thể tiếp theo tại vị trí t. Phương pháp thứ hai là chưng cất kiến thức ở cấp độ chuỗi (SeqKD)
[39]. SeqKD gợi ý một phương pháp đơn giản để chưng cất kiểu nhiệm vụ này, bằng cách thay thế văn bản thực tế y1···t bằng đầu ra tạo ra giáo viên yˆ1···t, còn được gọi là nhãn giả.

$$L(\theta) = \sum_{t=1}^{T} \alpha \log p(\hat{y}t+1 \mid \hat{y}1{:}t, x, \theta) + \beta \, KL\left[p(\cdot \mid \hat{y}1{:}t, x, \theta T) \,||\, p(\cdot \mid \hat{y}1{:}t, x, \theta)\right] \quad (2)$$

Tại đây θ là các tham số có thể đào tạo của mô hình sinh viên và α và β lần lượt kiểm soát trọng số của chuỗi và thuật ngữ mất từ.

Tối ưu hóa sở thích Giai đoạn thứ hai của việc điều chỉnh hướng dẫn cho LLM là sắp xếp chúng theo một tập hợp các sở thích của người dùng. Trong giai đoạn này, một tập hợp các cặp sở thích mong muốn được sử dụng để cải thiện đầu ra của mô hình. Mục tiêu là tạo ra các đầu ra y cho các lời nhắc x để tối đa hóa mô hình phần thưởng r trong khi vẫn duy trì gần với mô hình tham chiếu. Thông thường, mô hình tham chiếu được chọn là mô hình sau khi tinh chỉnh có giám sát. Đối với quá trình chưng cất, chúng ta có thể sử dụng giáo viên ban đầu một cách thuận tiện, tức là

$$\max_{\theta} \mathbb{E}_{x \sim D, y \sim p(y|x;\theta)} \, r_\varphi(x, y) - \beta KL\left[p(y \mid x; \theta) \,||\, \pi(y \mid x; \theta T)\right] \quad (3)$$

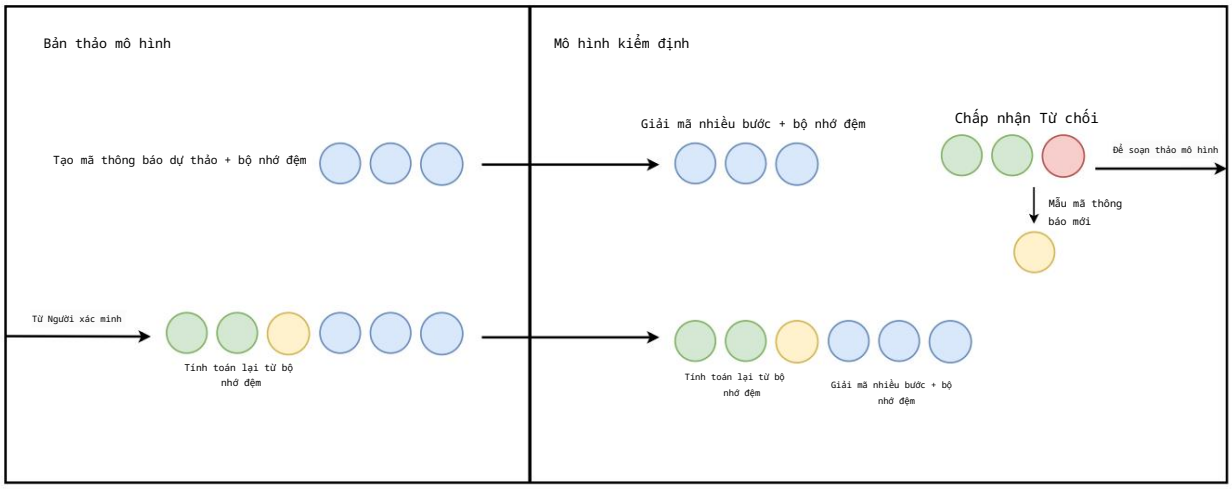
Mô hình sở thích này được xác định bởi hàm phần thưởng rφ(x, y) phụ thuộc vào phương pháp được sử dụng. Nghiên cứu trước đây sử dụng phản hồi AI chủ yếu tập trung vào việc sử dụng các phương pháp học tăng cường, chẳng hạn như tối ưu hóa chính sách gần (PPO) [64], để tối ưu hóa φ liên quan đến phần thưởng này. Gần đây, các phương pháp sử dụng tối ưu hóa sở thích trực tiếp (DPO) [58] đã có hiệu quả trong việc tối ưu hóa mục tiêu này với các bản cập nhật gradient trực tiếp. Cụ thể, DPO cho thấy rằng, nếu chúng ta có quyền truy cập vào các đầu ra yw được ưa thích và yl không được ưa thích cho một lời nhắc x nhất định, chúng ta có thể xây dựng lại vấn đề tối ưu hóa này như sau:

$$\pi\theta = \max_{\theta} \mathbb{E}_{(x,yw,yl) \sim D} \log \sigma\left(\beta \log \frac{p(yw|x; \theta)}{p(yw|x; \theta T)} - \beta \log \frac{p(yl|x; \theta)}{p(yl|x; \theta T)}\right). \quad (4)$$

Quá trình tối ưu hóa này có thể được thực hiện ở cấp độ trình tự bằng cách chấm điểm các đầu ra được ưa thích và không được ưa thích của mô hình với giáo viên và học sinh, sau đó truyền ngược lại cho học sinh. Theo như chúng tôi biết, đây là lần đầu tiên sử dụng DPO làm mục tiêu chưng cất.

4 Thuật toán giải mã suy đoán cho RNN tuyến tính

Mục tiêu chính của công thức RNN tuyến tính là cải thiện hiệu quả giải mã. Đối với cả RNN chú ý và tuyến tính, sự phụ thuộc tuần tự của thế hệ tự hồi quy vốn có làm giảm hiệu quả.
Hệ thống không thể sử dụng hết tất cả các tính toán có sẵn vì chúng cần phải đợi thế hệ các mã thông báo trước đó tiến hành [10, 11, 41, 67, 76]. Giải mã suy đoán đã nổi lên như một phương pháp phá vỡ nút thắt này bằng cách sử dụng thêm tính toán để suy đoán về các thế hệ tương lai. Trong phần này, chúng tôi xem xét các phương pháp áp dụng kỹ thuật này cho các mô hình Mamba lớn, sau đó có thể áp dụng cho các mô hình đã được tinh chế.



Hình 2: Giải mã suy đoán RNN nhiều bước. Trái (trên cùng): Mô hình dự thảo tạo ra tập hợp các mã thông báo dự thảo màu xanh theo trình tự. Sau đó, các mã thông báo dự thảo được xác minh. Phải (trên cùng): Xác minh sử dụng hạt nhân nhiều bước, mà không hiện thực hóa các trạng thái trung gian. Mã thông báo cuối cùng bị từ chối và thay thế bằng mã thông báo tốt nhất thực sự. Lưu ý rằng, ngay cả khi nhiều mã thông báo được tạo ra, chúng ta vẫn không thể tiến hành bộ đệm trạng thái ẩn. Trái (dưới cùng) Mô hình dự thảo hiện có thể tạo ra nhiều mã thông báo dự thảo màu xanh hơn từ các mã thông báo hiện tại, tạo ra tổng cộng sáu mã thông báo. Phải (dưới cùng) Khi dự thảo mới được xác minh, hạt nhân nhiều bước trả về cả trạng thái ẩn sau mã thông báo màu vàng và trạng thái ẩn cuối cùng, vì quá trình xác minh sẽ nằm giữa các vị trí đó.

4.1 Thách thức trong suy đoán RNN

Giải mã suy đoán sử dụng hai mô hình: mô hình bản nháp, θD và mô hình xác minh, θV . Mô hình bản nháp nhanh = $\arg\max_{y_{1:T}}$ tiềm năng trong tương lai, y kiểm tra xem chúng $p(y_1, \ldots, y_T ; \theta D)$, và mô hình xác minh lớn hơn tạo ra các hoàn thành có xếp hạng cao nhất tại mỗi bước thời gian hay không, tức là kiểm tra $p(y_t | y_{1:t-1} ; \theta V)$. Chuỗi càng dài trước khi xác minh lỗi thì đầu ra càng nhanh. Nếu một chuỗi một phần khớp, chúng ta có thể tua lại đến lần khớp cuối cùng.

Các mô hình dựa trên sự chú ý đặc biệt dễ bị suy đoán, vì chúng chậm khi tạo ra do bản chất tuần tự, nhưng nhanh khi xác minh do khả năng kiểm tra nhiều mã thông báo song song. Các mô hình RNN tuyến tính như Mamba có các đặc điểm hiệu suất khác biệt đáng kể khiến chúng ít dễ bị giải mã theo suy đoán hơn. Giải mã tuần tự sử dụng lấy mẫu theo kiểu hồi quy đã nhanh hơn đáng kể so với sự chú ý. Giống như sự chú ý, có các chế độ song song cho các mô hình như Mamba được sử dụng trong quá trình đào tạo. Những điều này hiệu quả, nhưng được điều chỉnh cho các chuỗi cực dài. Ngoài ra, chúng dựa vào các tối ưu hóa nhận thức phần cứng, chẳng hạn như tránh tạo ra các trạng thái trung gian. Các thuộc tính này khiến việc sử dụng để suy đoán cho các chuỗi tương đối ngắn trở nên khó khăn khi không biết khi nào xung đột sẽ xảy ra.

Một thách thức bổ sung phát sinh từ việc lưu trữ đệm trạng thái trong các mô hình RNN. Trạng thái của một mô hình chú ý được biểu diễn bằng bộ đệm khóa-giá trị, K1:t, V1:t; trong khi trạng thái của một mô hình RNN chỉ đơn giản là ht. Để cạnh tranh với sự chú ý, trạng thái RNN đơn lẻ này cần phải rất lớn. Trong quá trình suy đoán, chúng ta cần tua lại trạng thái trước đó tại bước thời gian t $V_{1:t'}$; tuy nhiên, đối với RNN, điều này sẽ yêu cầu lưu trữ đệm tất cả $h_{1:t'}$, điều này sẽ yêu cầu chi phí bộ nhớ lớn. Để chú ý, đây chỉ đơn giản là $K_{1:t'}$

4.2 Suy đoán RNN tuyến tính nhiều bước

Chúng tôi đề xuất một thuật toán mới cho giải mã suy đoán RNN tuyến tính bằng cách sử dụng thế hệ nhiều bước nhận biết phần cứng . Cốt lõi của hạt nhân thế hệ tiếp cận tính toán,

$$y_{j:k}, h_j , h_k \quad \text{MultiStep}(h_i , y_{1:n}, i, j, k; A, B, C, \ldots)$$

Trong đó i là trạng thái ẩn bắt đầu, $i \le j \le k$ và $j \ldots k$ là phạm vi đầu ra y cần thiết. Hạt nhân nhận thức được phần cứng vì nó tránh tạo ra các thuật ngữ chính từ bộ nhớ GPU nhanh. Cụ thể, nó tránh tạo ra hầu hết các tham số $h_{1:n}$ cũng như các tham số RNN tuyến tính theo thời gian rời rạc. Hạt nhân này nhằm mục đích nhắm mục tiêu

các vấn đề được trình bày ở trên. Cụ thể, nó có thể lưu ảnh chụp nhanh trạng thái hj trước khi đánh giá các mã thông báo dự thảo. Điều này cho phép tính toán lại trạng thái chính xác ngay sau khi một mã thông báo bị từ chối. Giả định là giải mã bị tắc nghẽn bởi bộ nhớ chứ không phải bởi tính toán, vì chúng ta có thể tính toán nhiều bước giải mã với rất ít tốn kém hơn so với giải mã một bước.

Thuật toán 2 và Hình 2 hiển thị thuật toán đầy đủ. Phương pháp này chỉ duy trì một trạng thái ẩn của RNN trong bộ nhớ đệm để xác minh và tiến hành nó một cách chậm chạp về sự thành công của hạt nhân nhiều bước. Kể từ khi các mô hình chưng cất chứa các lớp biến áp, chúng tôi cũng mở rộng giải mã suy đoán sang kiến trúc lai Attention/RNN. Trong cài đặt này, các lớp RNN thực hiện xác minh theo Thuật toán 2, trong khi các lớp biến áp chỉ thực hiện xác minh song song .

Lưu ý rằng nếu mô hình dự thảo là mô hình Mamba hoặc mô hình lai, phần suy đoán của thuật toán trở nên phức tạp hơn vì bản dự thảo mô hình cần tính toán lại trạng thái cho các mã thông báo được chấp nhận trong lần lặp trước. Điều này được thực hiện tương tự như mô hình xác minh, bằng cách lưu trữ các mục cũ hơn và tính toán lại trong vòng suy đoán tiếp theo.

Thuật toán 2 Suy đoán RNN tuyến tính nhiều bước

```
hàm Verify(y1:k, j, hi)
    // y1:k là bản nháp, j là bản xác minh cuối cùng,
    //,hi là trạng thái được lưu trong bộ nhớ đệm với i ≤ j
    và y:k , hj , hk   MultiStep(hi , y1:k, i, j, k; θv)
    tôi    Xung đột đầu tiên(yj:k, y′ j:k)
    trả về k   ,hk nếu k  = k khác hj

hàm Speculate(K)
    // K mã thông báo được soạn thảo cho mỗi bước
    hcache   h0
    j   0
    trong khi yj không phải là kết thúc
        k   j + K
        yj+1:k   đối số tối đa p(yj+1:k | y1:j , θD)
        j, hcache   Xác minh(y1:k, j, hcache)
    trả về y1:j
```

4.3 Phân tích suy đoán và phần cứng Tối ưu hóa cụ thể

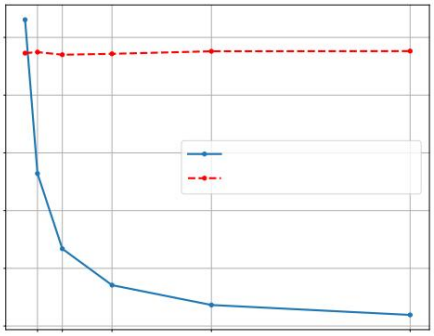
Để xác minh hiệu quả của phương pháp này, chúng tôi chạy suy đoán bằng cách sử dụng Mamba 7B và Mamba 2.8B làm mô hình mục tiêu. Kết quả được thể hiện trong Bảng 1. Hình 3 cho thấy các đặc điểm hiệu suất của Multi-Step chính hạt nhân.

Kích thước mô hình	GPU	K	# Gen. Token	Thông lượng (toks/giây)	Tăng tốc
2,8 tỷ	3090	3	3.01	259	2,3 lần
2,8 tỷ	3090	4	3.28	289	2,6 lần
2,8 tỷ	H100	3	4.04	389	1,71x
2,8 tỷ	H100	4	3.9	421	1,85 lần
7B	3090	3	3.19	109	2,1 lần
7B	3090	4	3,56	110	2,1 lần
7B	H100	3	3.28	271	1,95 lần
7B	H100	4	3.6	272	2 lần

Bảng 1: Kết quả tăng tốc cho giải mã suy đoán với các mô hình Mamba thuần túy. Bộ xác minh 2.8B sử dụng 130M Bản nháp Mamba. Trình xác minh 7B sử dụng bản nháp Llama3 1B mà chúng tôi đã đào tạo. Dữ liệu từ The Pile. K là số bản nháp mã thông báo được tạo ra, # Gen bao gồm một mã thông báo bổ sung từ nhật ký xác minh cuối cùng.

Tăng tốc trên GPU H100. Một triển khai ngây thơ của thuật toán của chúng tôi đã cho thấy hiệu suất mạnh mẽ trên GPU Ampere như được hiển thị trong Bảng 1. Tuy nhiên, đạt được hiệu suất mạnh mẽ trên GPU H100 thì tốt hơn nhiều đầy thách thức. Điều này chủ yếu là do các hoạt động của GEMM diễn ra nhanh hơn nhiều, khiến chi phí phát sinh từ các hoạt động lưu trữ đệm và tính toán lại dễ thấy hơn. Trong thực tế, việc triển khai ngây thơ của chúng tôi thuật toán, với một số lệnh gọi kernel khác nhau, đạt được tốc độ tăng đáng kể trên 3090 GPU (1,5 lần đối với Mamba 2.8B với tỷ lệ chấp nhận 60%) nhưng không tăng tốc chút nào trên H100.

Chúng tôi đã tối ưu hóa việc triển khai của mình bằng cách hợp nhất các hạt nhân và bằng cách điều chỉnh việc triển khai để dễ dàng cho phép lưu trữ đệm và tính toán lại các bước cũ. Cụ thể, mô hình xác minh thực hiện i) tính toán lại các bước trước đó từ bộ nhớ đệm, ii) giải mã nhiều bước cho chuỗi mã thông báo dự thảo mới và iii) lưu trữ đệm trong một hạt nhân duy nhất 1. Đối với mô hình dự thảo, tính toán lại, giải mã và lưu trữ đệm cũng được hợp nhất trong một hạt nhân duy nhất. Các triển khai kết quả lưu trữ tốc độ tăng trên GPU H100s, như thể hiện trong Bảng 1.



Hình 3: Hiệu suất của hạt nhân SSM nhiều bước để tạo ra 32 mã thông báo.

5 Kết quả

5.1 Thiết lập thử nghiệm

Các mô hình mục tiêu. Chúng tôi thực hiện các thí nghiệm bằng cách sử dụng hai mô hình trò chuyện LLM: Zephyr-7B [72], là một trò chuyện được tinh chỉnh Mistral 7B [37], Llama-3 Instruct 8B [21]. Đối với các mô hình RNN tuyến tính, chúng tôi sử dụng các phiên bản lai của Mamba và Mamba2 với các lớp chú ý 50%, 25%, 12,5% và 0%. Chúng tôi gọi 0% là mô hình Mamba thuần túy.

Mamba2 là một biến thể kiến trúc của Mamba được thiết kế để hướng đến các kiến trúc GPU gần đây hơn.

Zephyr-Mamba đề cập đến quá trình chưng cất từ Zephyr [72], trong khi Llama3-Mamba / Llama3-Mamba2 chỉ ra quá trình chưng cất từ Llama-3 hướng dẫn 8B [71]. Nói một cách chính xác, Mamba-Zephyr chưng cất của chúng tôi là một mô hình dưới bốn, vì Zephyr/ Mistral-8B sử dụng kiến trúc chú ý cửa sổ trượt. Mamba-Zephyr chưng cất của chúng tôi (50%) có kiến trúc tương tự như Samba [60].

Đào tạo. Chưng cất không yêu cầu bất kỳ dữ liệu tiền đào tạo mô hình ngôn ngữ nào, mà thay vào đó sử dụng quy trình đào tạo sau để điều chỉnh mô hình mới. Chúng tôi sử dụng quy trình ba giai đoạn. Ở giai đoạn đầu tiên, chúng tôi sử dụng UltraChat [20] và UltraFeedback [17] làm lời nhắc hạt giống và sử dụng mô hình giáo viên để tạo nhãn giả.

Mô hình học sinh được đào tạo trong một kỷ nguyên bằng cách sử dụng mất mát L trong Eq 2 với $\alpha = 1$ và $\beta = 0{,}1$. Các mô hình được đào tạo bằng trình tối ưu hóa AdamW với $\beta = (0{,}9, 0{,}98)$ với kích thước lô 64. Chúng tôi sử dụng khởi động tốc độ học tuyến tính (cho 500 bước đầu tiên) theo sau là ủ cosin. Ở giai đoạn thứ hai, chúng tôi sử dụng điều chỉnh có giám sát với mô hình của mình trên các tập dữ liệu GenQA [12], InfinityInstruct [3] và OpenHermes 2.5 [70] bằng cách sử dụng SFT trong một kỷ nguyên, với các siêu tham số giống như Zephyr [72]. Ở giai đoạn cuối cùng, đối với các mô hình được chưng cất từ Zephyr, chúng tôi thực hiện căn chỉnh chưng cất với mô hình của mình bằng cách sử dụng DPO trên tập dữ liệu UltraFeedback [17] phù hợp với mô hình giáo viên. Trong khi các mô hình được chưng cất từ Llama-3 được hướng dẫn 8B, chúng tôi sử dụng các tập dữ liệu từ SimPO [51] và Zephyr [72]. Chúng tôi chỉ đông lạnh Gated MLP (FFN) ở giai đoạn đầu tiên, trong khi ở giai đoạn thứ hai và cũng là giai đoạn cuối cùng, tất cả các thông số đều được đào tạo 2. Toàn bộ quá trình chưng cất cho mỗi mô hình lai (ví dụ: Mamba-Llama3 (50% att)) mất chưa đầy năm ngày trong 8x80G A100.

Đường cơ sở. Ngoài các kiến trúc Transformer cốt lõi, các đường cơ sở chính mà chúng tôi so sánh là các mô hình RNN tuyến tính quy mô lớn khác. Chúng tôi so sánh với cả kiến trúc SSM thuần túy, chẳng hạn như TRI Mamba 7B [52] được đào tạo với 1,2T mã thông báo và Falcon Mamba 7B3 được đào tạo với hơn 5T mã thông báo, kiến trúc SSM lai, chẳng hạn như Nvidia Hybrid Mamba 2 [74] được đào tạo với 3,7T mã thông báo và các mô hình RNN lai tuyến tính khác, chẳng hạn như Recurrent Gemma-9B Instruct [8, 19].

Sau khi phát hành các mô hình biến đổi SoTA mới ở thang đo 8B và 3B, Llama-3.1 và Llama-3.2, chúng tôi đã hợp lý hóa quy trình chưng cất và hiện đang chưng cất bằng mô hình giáo viên Llama-3.1 70B lớn hơn trong khi khởi tạo các mô hình có thang đo 3B và 8B có kích thước tương tự. Chúng tôi chưng cất mô hình của mình trên các tập dữ liệu GenQA [12] và InfinityInstruct [3] , tạo ra Mamba-Llama3.2-3B, Mamba2-Llama3.2-3B, Mamba-Llama3.1-8B và Mamba2-Llama3.1-8B. Ngoài ra, chúng tôi thực hiện thêm DPO trên các mô hình này bằng cách sử dụng cùng một tập dữ liệu như trước, tạo ra Mamba-Llama3.2-3B-dpo, Mamba2-Llama3.2-3B-dpo, Mamba-Llama3.1-8B-dpo và Mamba2-Llama3.1-8B-dpo. Giai đoạn chưng cất mất tám ngày trên 8xA100 và bốn ngày trên 8xH100.

1Ngoài ra, chúng tôi triển khai phần tích chập của khối Mamba bằng cách sử dụng bộ đệm tròn cho phép chúng tôi theo dõi các mục cũ và đưa chúng vào tích chập khi cần để tính toán lại.

2Chúng tôi đóng băng các lớp MLP ở giai đoạn đầu tiên vì chúng tôi muốn tạo ra một mô hình tương tự như mô hình khởi tạo. Tuy nhiên, trong quá trình chưng cất đầu cuối, chúng tôi chỉ tập trung vào tổn thất KL, do đó, việc đào tạo tất cả các tham số (không đóng băng các lớp MLP) sẽ mang lại kết quả tốt hơn.

3https://huggingface.co/tiiuae/falcon-mamba-7b

5.2 Đánh giá trên các tiêu chuẩn trò chuyện

Chúng tôi đánh giá các mô hình của mình bằng cả chuẩn mực trò chuyện một lượt, AlpacaEval [43] và nhiều lượt, MT-Bench [84]. Các chuẩn mực này đánh giá khả năng của mô hình trong việc tuân theo hướng dẫn và phản hồi với các thách thức lời nhắc trên nhiều lĩnh vực khác nhau.

Mô hình (% Att)	Căn chỉnh kích thước	MT-Bàn (điểm)	MT-Bàn (Vòng 1)	MT-Bàn (Vòng 2)	AlpacaĐánh giá (Tỷ lệ thắng LC %)	AlpacaĐánh giá (thắng %)
Zephyr	7B DPO	7.34	-	-	13.200.96	10.990.96
Mamba-Zephyr (50%)	7B DPO Mamba-	7.31	-	-	20.660.74	16.691.10
Zephyr (25%)	7B DPO Mamba-	7.03	-	-	17.160.69	13.111.00
Zephyr (12,5%) 7B DPO		6.40	-	-	15.320.66	12.961.02
Llama-3.1-Chỉ dẫn 8B RLHF Mamba-Llama3.1 (50%)		8.0	-	-	20,9	21,8
8B Mamba2-Llama3.1 (50%) 8B Mamba-		7.7	8.0	7.3	18,971,23	21.221.23
Llama3.2 (50%) 3B Mamba2-Llama3.2		7.6	8.1	7.0	18,991,24	21.551.24
(50%) 3B		6.9	7.6	6.1	13,571,08	15.541.08
		6,5	7.1	5.8	12,611,05	14.341.05
Llama-3-Hướng dẫn	8B RLHF	8,00	-	-	22.901.26	22.601.26
Mamba-Llama3 (50%)	8B DPO Mamba-	7,35	7,82	6,88	29.611.31	26.691.31
Llama3 (25%)	8B DPO Mamba-	6,86	7,56	6,15	25.851.26	22.501.26
Llama3 (12,5%) 8B DPO		6,46	6,91	6,01	20.761.16	17.931.16
Mamba2-Llama3 (50%) 8B DPO Mamba2-Llama3		7,32	7,93	6,70	26.781.26	22.691.26
(25%) 8B DPO Mamba2-Llama3 (12,5%) 8B DPO		6,74	7,24	6,24	22.751.18	19.011.18
Mamba2-Llama3 (0%)		6,48	6,83	6,13	20.251.13	16.881.13
	8B DPO	5,64	6.16	5.11	14.490.93	10.880.93
Hướng dẫn Falcon Mamba	7B SFT	6,40	7,25	5,55	4.040.45	2.150.45
GPT-3.5-turbo	- RLHF	7,94	-	-	22,70	14.10
GPT-4o	- RLHF	-	-	-	57.461.47	51.331.47

Bảng 2: Kết quả đánh giá chuẩn trò chuyện cho các mô hình truy cập mở và độc quyền trên MT-Bench và AlpacaEval. MT-Bench chấm điểm mô hình phản hồi bằng cách sử dụng GPT-4. AlpacaEval phiên bản hai đo tỷ lệ thắng-thua giữa mô hình cơ sở và GPT-4 được chấm điểm bởi GPT-4 Turbo.

Bảng 2 cho thấy hiệu suất của các mô hình của chúng tôi trên các điểm chuẩn trò chuyện so với máy biến áp lớn mô hình. Mô hình Mamba lai chưng cất (50%) đạt được điểm số tương tự trong chuẩn MT như mô hình giáo viên và tốt hơn một chút so với mô hình giáo viên trên chuẩn mực AlpacaEval trong cả hai chiến thắng LC tỷ lệ và tỷ lệ thắng chung. Hiệu suất của Mamba lai chưng cất (25% và 12,5%) kém hơn một chút so với của các mô hình giáo viên trong chuẩn mực MT nhưng vẫn vượt trội hơn một số máy biến áp lớn ngay cả với nhiều hơn tham số trong AlpacaEval. Mô hình tinh khiết chưng cất (0%) giảm đáng kể về độ chính xác. Đáng chú ý là mô hình lai chưng cất hoạt động tốt hơn Falcon Mamba, được đào tạo từ đầu với nhiều hơn hơn 5T token.

5.3 Đánh giá trên các chuẩn mực chung

Đánh giá Zero Shot. Chúng tôi sử dụng thư viện LM Evaluation Harness mã nguồn mở [25] (nhánh big-refactor) để đánh giá 10 nhiệm vụ, với các số liệu đánh giá sau: Độ chính xác của WinoGrande (WG) [62], PIQA (PQ) độ chính xác [7], độ chính xác được chuẩn hóa của HellaSwag (HS) [82], độ chính xác của ARC-Easy và ARC-Challenge (AE và AC) và độ chính xác được chuẩn hóa, [14], MMLU (MM), độ chính xác [33], độ chính xác được chuẩn hóa của OpenBookQA (OB) [54], Độ chính xác TruthFulQA (TQ) [46], độ chính xác PubMedQA (PM) [38] và độ chính xác RACE (RA) [40]. Mỗi nhiệm vụ là được đánh giá bằng cách phân tích xác suất được mô hình gán cho mỗi lựa chọn câu trả lời tiềm năng.

Bảng 3 cho thấy đánh giá không có phát bắn nào trong chuẩn mực LM Eval cho Mamba và Mamba2 được chưng cất từ các nguồn khác nhau mô hình giáo viên. Cả hai mô hình lai Mamba-Llama3 và Mamba2-Llama3, được chắt lọc từ Llama-3 Hướng dẫn 8B, hoạt động tốt hơn so với các mô hình TRI Mamba và Nvidia Mamba mã nguồn mở được đào tạo

Mô hình (% Att)	WG PI HS AE AC MM OB TQ PM RA TRUNG BÌ NH
TRI Mamba-7B	71,42 81,01 77,93 77,53 46,67 33,39 46,20 32,09 72,30 37,99 57,65
Nvidia Hybrid Mamba-8B	71,27 79,65 77,68 77,23 47,70 51,46 42,80 38,72 69,80 39,71 59,60
Llama-3.1-8B-Hướng dẫn	73,88 80,79 79,21 81,78 55,20 68,12 43,20 42,67 75,20 44,78 64,48
Llama3.1-Mamba (50%)	72,77 79,33 75,91 82,24 53,84 62,13 42,80 40,02 72,00 42,11 62,32
Llama3.1-Mamba-DPO (50%)	73,80 80,41 77,36 84,01 56,57 63,50 44,20 46,07 74,40 43,44 64,38
Llama3.1-Mamba2 (50%) 71,74 78,89 75,36 82,20 52,65 61,01 41,60 40,31 72,60 42,11 61,85	
Llama3.1-Mamba2-DPO (50%) 74,11 80,03 79,69 84,81 59,73 59,74 44,00 50,22 74,60 46,12 65,31	
Llama-3.2-3B-Hướng dẫn 67,48 75,68 70,43 74,07 45,90 60,43 36,00 38,01 69,60 40,67 57,83	
Llama3.2-Mamba (50%) 67,32 77,31 70,37 77,65 48,38 54,48 39,40 42,02 66,40 40,29 58,36	
Llama3.2-Mamba-DPO (50%) 67,40 77,31 72,56 79,97 52,65 55,09 41,60 48,53 70,00 43,64 60,88	
Llama3.2-Mamba2 (50%) 66,06 76,01 69,13 76,68 46,67 53,12 38,80 34,78 63,80 39,81 56,49	
Llama3.2-Mamba2-DPO (50%) 67,32 77,69 74,45 80,26 54,10 52,47 42,40 50,28 65,40 43,44 60,78	
Mamba-Zephyr (50%) 68,82 80,36 76,91 81,40 55,63 55,43 42,60 41,99 72,60 42,20 61,79	
Mamba-Llama3 (50%)	68,98 78,02 78,43 74,45 51,96 57,81 44,00 47,69 73,00 38,56 61,30
Mamba-Llama3 (25%)	62,83 78,07 75,00 74,28 47,35 53,50 40,00 43,64 65,40 36,94 57,70
Mamba-Llama3 (12,5%)	59,75 75,08 71,71 70,58 43,60 49,81 41,40 41,41 62,40 34,45 55,02
Mamba2-Llama3 (50%)	71,51 81,45 79,47 78,83 58,19 55,70 44,20 57,74 72,4 38,85 63,84
Mamba2-Llama3 (25%)	64,80 78,73 77,7 76,35 52,47 53,71 42,40 55,33 64,80 39,23 60,55
Mamba2-Llama3 (12,5%)	63,38 76,82 73,14 75,84 50,26 50,78 39,60 50,00 65,80 36,46 58,21
Mamba2-Llama3 (0%)	58,56 76,82 70,75 74,12 47,95 45,19 39,00 40,20 62,20 32,63 54,74

Bảng 3: Đánh giá trên chuẩn LM Eval cho Mamba và Mamba2 được chưng cất từ Llama-3 Instruct 8B.

từ đầu. Hiệu suất giảm xuống với nhiều lớp RNN tuyến tính hơn, nhưng vẫn có khả năng cạnh tranh ở mức 25% so với các mô hình được đào tạo từ đầu.

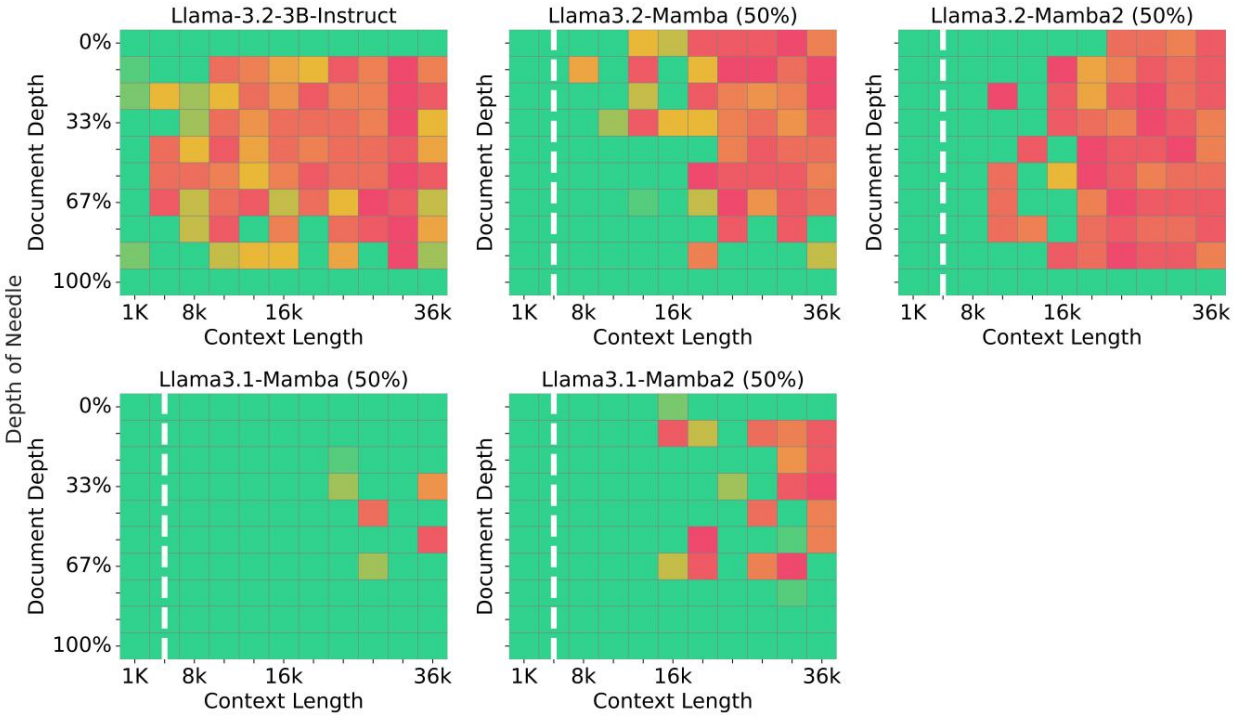
Đánh giá chuẩn. Chúng tôi cũng báo cáo các đánh giá ít lần trên OpenLLMLeaderboard bằng cách tiến hành 25 phát bắn vào ARC-Challenge [15], 10 phát bắn vào HellaSwag [82], 5 phát bắn vào MMLU [34] và 5 phát bắn vào Winogrande [62]. Đối với TruthFulQA, số liệu mc2 được báo cáo trong chuẩn mực này. Đối với GSM8K [16], chúng tôi theo dõi đánh giá để hướng dẫn mô hình điều chỉnh [51], sử dụng ZeroEval [45], một chuẩn mực được thiết kế cho trò chuyện mô hình. Chúng tôi cũng bao gồm CRUX [29] từ chuẩn mực đó, được thiết kế để đánh giá lý luận về mã. Tất cả các mô hình được đánh giá bằng giải mã tham lam trong ZeroEval.

Mô hình (% Att)	ARC HS MMLU WG TQ GSM8K CRUX				
Chim ưng Mamba-7B	62,03 80,82	62,11	73,64 53,42	41,32	8,88
Gemma-9B tái phát	52,00 80,40	60,50	73,60 38,60	38,51	26,25
Mamba-Llama3 (50%)	56,57 78,99	59,26	69,06 58,85	67,85	27,88
Mamba-Llama3 (25%)	55,03 75,66	52,68	62,83 55,03	40,64	15.62
Mamba-Llama3 (12,5%)	52,90 72,46	49,20 59,19 53,00		26,91	11.25
Mamba2-Llama3 (50%)	60,41 77,97	56,67	71,35 66,60	59,36	24,88
Mamba2-Llama3 (25%)	59,22 76,88	53,94	64,88 64,64	38.13	13,25
Mamba2-Llama3 (12,5%) 53,33	72,16 Mamba2-	50,85	63,61 61,12	35.03	10,25
Llama3 (0%) 53,51 70,31		44,21	58,91 52,31	-	-

Bảng 4: Kết quả trên Bảng xếp hạng Open LLM và Bảng xếp hạng ZeroEval. Đối với GSM8K và CRUX, chúng tôi đã chọn đánh giá zero-shot sử dụng ZeroEval, được thiết kế để đánh giá các mô hình hướng dẫn. Chúng tôi đã đánh giá các mô hình được điều chỉnh theo hướng dẫn tương ứng cho Falcon Mamba-7b và RecurrentGemma-9B, cụ thể là Falcon Mamba-7b-instruct và RecurrentGemma-9B-it.

Bảng 4 cho thấy hiệu suất của các mô hình lai chưng cất của chúng tôi phù hợp với hiệu suất của các mô hình nguồn mở tốt nhất các mô hình RNN tuyến tính trên Bảng xếp hạng LLM mở, đồng thời vượt trội hơn các mô hình nguồn mở tương ứng của chúng hướng dẫn các mô hình trong GSM8K và CRUX.

5.4 Đánh giá về các nhiệm vụ ngữ cảnh dài



Hình 4: Đánh giá Needle in a Haystack. Các ô vuông màu xanh lá cây biểu thị tỷ lệ thành công truy xuất cao, trong khi đường đứt nét màu trắng đánh dấu các ví dụ dài nhất gặp phải trong quá trình đào tạo chưng cất. Trục Y biểu thị khoảng cách đến mục tiêu đã truy xuất.

Hình 4 minh họa kết quả của Needle in a Haystack. Mặc dù độ dài chưng cất chỉ là 2k, các mô hình 3B chưng cất của chúng tôi (Mamba-Llama3.2-3B (50%) và Mamba2-Llama3.2-3B (50%)) đạt được độ chính xác hoàn hảo lên đến 10k, tốt hơn Llama-3.2-3B-Instruct. Tương tự như vậy, các mô hình 8B chưng cất (Mamba-Llama3.1-8B (50%) và Mamba2-Llama3.1-8B (50%)) đạt được độ chính xác hoàn hảo lên đến 16k, với Mamba-Llama3.1-8B cho thấy kết quả tốt lên đến 38k.

5.5 Thiết lập giải mã suy đoán lai Chúng

tôi thực hiện giải mã suy đoán bằng cách sử dụng các mô hình lai đã chưng cất. Chúng tôi chạy thử nghiệm bằng cách sử dụng cả Hybrid Mamba 50% và Hybrid Mamba 25% làm mô hình chính. Đối với các mô hình dự thảo, chúng tôi đào tạo các mô hình Transformer Draft 2 và 4 lớp trên tập dữ liệu OpenHermes2.5 [70], trong khoảng 3 kỷ nguyên đầy đủ, theo phương pháp "thu nhỏ và tinh chỉnh" từ [66]. Cụ thể, chúng tôi khởi tạo các lớp dự thảo bằng cách sử dụng các lớp từ mô hình Zephyr-7B (chúng tôi lấy các lớp ở chỉ số [0, 31] cho mô hình 2 lớp và [0, 10, 20, 31] cho mô hình 4 lớp) và các nhúng và mô hình ngôn ngữ cũng từ mô hình Zephyr-7B [72]. Chúng tôi thực hiện che giấu mất mát trên lời nhắc, do đó chỉ xem xét mất mát dự đoán mã thông báo tiếp theo (entropy chéo) trên các phần tiếp tục trò chuyện từ tập huấn luyện. Các thí nghiệm giải mã suy đoán được chạy trên một NVIDIA RTX 3090 duy nhất dựa trên dữ liệu từ OpenHermes2.5.

Bảng kết quả 5 cho thấy kết quả giải mã suy đoán lai ghép, sử dụng cả mô hình lai Zephyr và Llama với các cấu hình khác nhau. Đối với cả mô hình chưng cất 50% và 25% , chúng tôi đạt được tốc độ tăng hơn 1,8 lần trên Zephyr-Hybrid so với đường cơ sở không suy đoán. Chúng tôi cũng chỉ ra rằng mô hình bản nháp 4 lớp mà chúng tôi đã đào tạo đạt được tỷ lệ chấp nhận cao hơn, nhưng nó làm tăng thêm một số chi phí do kích thước mô hình bản nháp tăng lên. Đối với mô hình lai Llama, tốc độ tăng khiêm tốn hơn vì bản nháp

Mô hình dự thảo K	Mô hình mục tiêu (% Att)	# Tăng tốc mã thông báo chung	
2 lớp	4 Mamba-Zephyr (50%)	2,48	1,8 lần
	4 Mamba-Zephyr (25%)	2,64	1,88 lần
4 lớp	4 Mamba-Zephyr (50%)		1,81x
	4 Mamba-Zephyr (25%)	3 3	1,8 lần
4 lớp 4 lớp	3 Mamba-Llama3 (50%)	2.7	1,6 lần
	4 Mamba-Llama3 (50%)	3.6	1,58 lần

Bảng 5: Các số liệu hiệu suất cho các cấu hình mô hình mục tiêu và dự thảo khác nhau cho K = 4 trên dữ liệu từ OpenHermes2.5. # Gen là số lượng trung bình các mã thông báo được tạo ra trên mỗi bước giải mã suy đoán và bao gồm một mã thông báo bổ sung từ nhật ký xác minh cuối cùng.

mô hình lớn hơn do bảng nhúng lớn của Llama 3. Trong công việc tiếp theo, chúng tôi sẽ tập trung vào việc tạo những mô hình dự thảo này nhỏ hơn.

6 Phân tích

Mô hình (% Att)	Tỷ lệ PPL	
Giáo viên: Zephyr (7B)	2.02	1
Mamba-Zephyr (50%)	2.09	1.03
Mamba-Zephyr (25%)	2.20	1.09
Mamba-Zephyr (6,25%) 2,46 Mamba-Zephyr (0%)	3.36	1.22 1,66
Giáo viên: Pythia (70M)	51,4	1
Chưng cất linh cẩu	121,2 2,36	

Người mẫu	Mamba Hyb Mamba Hyb Mamba (50% chú ý)	(25% chú ý)
Dis	5,55	5.01
Dis+SFT	5,61	4,97
Dis+DPO	5,42	4,84
Dis+SFT+DPO	6,69	6.10

Bảng 6: (Trái) So sánh sự phức tạp giữa phương pháp chưng cất của chúng tôi và [59]. (Phải) Nghiên cứu cắt bỏ các phương pháp căn chỉnh khác nhau của Distilled Hybrid Mamba trên MT-benchmark sử dụng OpenHermes 2.5 như tập dữ liệu SFT.

So sánh với các phương pháp chưng cất khác Bảng 6 (bên trái) so sánh sự phức tạp của các mô hình khác nhau các biến thể. Chúng tôi chưng cất bằng cách sử dụng Ultrachat như lời nhắc hạt giống [20] trong một thời đại và so sánh sự bối rối. Chúng tôi thấy rằng việc loại bỏ nhiều lớp hơn sẽ tệ hơn đáng kể. Chúng tôi cũng so sánh phương pháp chưng cất của chúng tôi với một đường cơ sở trước đó. Cách tiếp cận này chất lọc một mô hình Transformer thành một mô hình Hyena [57], như đã đề xuất trong [59]. Họ sử dụng một phương pháp chưng cất khác bằng cách sử dụng chuyển giao kiến thức tiến bộ, trong đó học sinh mô hình được đào tạo bắt đầu từ lớp đầu tiên và dần dần mở rộng sang các lớp tiếp theo. Trong khi nó rất khó để so sánh, quá trình chưng cất của chúng tôi cho thấy sự suy thoái nhỏ hơn (1,03 đối với 50% sự chú ý, 1,09 đối với 25% chú ý, 1,22 cho 6,35% chú ý và 3,36 cho không chú ý), trong khi mô hình Distill Hyena được đào tạo trong Bộ dữ liệu WikiText [53] có mô hình nhỏ hơn nhiều và cho thấy sự suy giảm độ phức tạp lớn.

Liệu việc chưng cất từ sở thích có giúp ích không? Trong Bảng 6 (Bên phải), chúng tôi trình bày tác động của các bước khác nhau trong quá trình căn chỉnh của quá trình chưng cất. Chúng tôi quan sát thấy rằng SFT hoặc DPO riêng lẻ không mang lại nhiều cải thiện, trong khi SFT + DPO mang lại điểm số tốt nhất. Các mô hình được đào tạo bằng cách sử dụng Zephyr làm mô hình giáo viên và Bộ dữ liệu OpenHermes 2.5 [70] là bộ dữ liệu SFT và UltraFeedback [17] là bộ dữ liệu DPO.

Phá hủy chưng cất nhãn giả. Chúng tôi xem xét một số nghiên cứu phá hủy mô hình khác nhau trong Bảng 7. Đối với những thí nghiệm này chúng tôi xem xét việc đào tạo cho 5k bước bằng cách sử dụng các phương pháp tiếp cận nhãn giả trên Ultrachat [20] tập dữ liệu. Bảng 7 (Bên trái) trình bày kết quả chưng cất với nhiều khởi tạo khác nhau. Theo đó bảng, việc khởi tạo trọng số từ một máy biến áp là rất quan trọng đối với hiệu suất. Nếu không có việc khởi tạo trọng số từ một máy biến áp, sự bối rối trở nên tồi tệ hơn đáng kể đối với cả mô hình Mamba thuần túy và mô hình lai. Ngoài ra,

Người mẫu	Ba (0% Áp dụng) Đông lạnh - Đông lạnh	Mamba Hyb (50% chú ý) Đông lạnh - Đông lạnh
+ Chú ý-Init	3.36 66.7	2.09 -Chú ý-Init 9.1
	18.2 20.3	7.4 11.2

Người mẫu	Mamba Hyb (25% chú ý) Bước - Bước	Mamba Hyb (50% chú ý) Bước - Bước
+ Xen kẽ 2.20	- 2.29	2.09 -
Xen kẽ 2.89	-	2.41 -

Bảng 7: (Trái) So sánh độ phức tạp với các khởi tạo khác nhau ở giai đoạn đầu. (Phải) So sánh độ phức tạp với các lớp xen kẽ Mamba khác nhau và chưng cất từng bước ở giai đoạn đầu tiên.

đóng băng các lớp MLP có thể giúp mô hình học sinh tập trung vào việc học tương tác của các mã thông báo và mô phỏng tốt hơn các lớp chú ý. Bảng 7 (Bên phải) cũng cho thấy lợi ích nhỏ hơn từ chưng cất và xen kẽ tiến bộ sự chú ý với Mamba.

Chú ý Khởi tạo. Chúng tôi so sánh khởi tạo ngẫu nhiên mặc định của Mamba với việc sử dụng lại tuyến tính chiếu từ sự chú ý sử dụng cùng một công thức. Cả hai mô hình đều được đào tạo bằng cách sử dụng Zephyr làm giáo viên mô hình và tập dữ liệu OpenHermes 2.5 [70] là tập dữ liệu SFT và UltraFeedback [17] là tập dữ liệu DPO.

Người mẫu	LAMBADA (mọi người)	MMLU	ARC-C	TruthfulQA	HellaSwag	MT-Bàn (điểm)	AlpacaĐánh giá (Tỷ lệ thắng LC %)
+ Chú ý init	47,98	49,15	46,67	75.07	6,69	14.11	
- Chú ý init	6.20 55.01	26.21	25,26	34.01	27,91	1.04	0,02

Bảng 8: Hiệu suất của Zephyr-Mamba (chú ý 50%) với các khởi tạo khác nhau.

Bảng 8 so sánh hiệu suất của mô hình lai sử dụng hai phương pháp khởi tạo khác nhau: mặc định khởi tạo ngẫu nhiên và tái sử dụng phép chiếu tuyến tính từ sự chú ý. Mô hình thực hiện đáng kể tốt hơn với việc tái sử dụng phép chiếu tuyến tính từ sự chú ý so với khởi tạo ngẫu nhiên, trên tất cả đánh giá chuẩn mực. Kết quả này xác nhận rằng việc khởi tạo từ trọng số chú ý là rất quan trọng.

Người mẫu	LAMBADA (mọi người)	MMLU	ARC-C	TruthfulQA	HellaSwag	MT-Bàn (điểm)	AlpacaĐánh giá (Tỷ lệ thắng LC %)
50% Att w/o Mamba	6.20	47,98	49,15	46,67	75.07	6,69	14.11
50% Att w/o Mamba 151,98		24,46	21,93	32,39	27,91	1.01	0

Bảng 9: Hiệu suất của Hybrid-Mamba với các khởi tạo khác nhau.

Sự cần thiết của RNN tuyến tính. Chúng tôi đào tạo một mô hình loại bỏ hoàn toàn các khối Mamba khỏi mô hình bằng cách sử dụng cùng một công thức để xem mô hình có thể thích ứng hay không. Cả hai mô hình đều được đào tạo bằng cách sử dụng Zephyr làm mô hình giáo viên, với tập dữ liệu OpenHermes 2.5 [70] là tập dữ liệu SFT và UltraFeedback [17] là tập dữ liệu DPO. Bảng 9 so sánh hiệu suất của mô hình có và không có khối Mamba. Mô hình có Mamba thực hiện tốt hơn đáng kể so với cái không có nó. Điều này khẳng định rằng việc thêm các lớp Mamba là rất quan trọng và hiệu suất được cải thiện không chỉ đơn thuần là nhờ vào cơ chế chú ý còn lại.

7 Công trình liên quan

Các mô hình không chú ý. Các mô hình không chú ý cung cấp hiệu quả tính toán và bộ nhớ được cải thiện, giúp Chúng ngày càng phổ biến cho nhiều tác vụ xử lý ngôn ngữ khác nhau, bao gồm cả mô hình hóa ngôn ngữ hồi quy tự động. Các mô hình như S4 [28] và các biến thể tiếp theo của nó [27, 31] đã cho thấy kết quả đầy hứa hẹn trong tổng hợp tầm xa nhiệm vụ [69]. Kiến trúc SSM có cổng, chẳng hạn như GSS [50] và BiGS [75], kết hợp cơ chế cổng vào SSM cho mô hình ngôn ngữ (hai chiều). Mô hình Mamba mới được giới thiệu [26] lập luận rằng động lực tĩnh của các phương pháp này không kết hợp được lựa chọn ngữ cảnh cụ thể cho đầu vào trong trạng thái ẩn,

có thể rất quan trọng đối với các nhiệm vụ như mô hình hóa ngôn ngữ. Mamba đã được chứng minh là vượt trội hơn Transformers trên các kích thước và quy mô mô hình khác nhau. Ngoài ra, một số kiến trúc mô hình dưới bậc hai khác [1, 2, 4, 19, 22, 57, 79, 80] và kiến trúc lai [23, 44] cũng đã được đề xuất.

Chưng cất từ Transformers. Có tương đối ít nỗ lực để chưng cất thành các mô hình kiểu RNN tuyến tính. Laughing Hyena [49] đề xuất chưng cất tích chập dài thành biểu diễn không gian trạng thái, cho phép suy luận thời gian không đổi trong Hyena [57]. Ralambomihanta et al. [59] giới thiệu một phương pháp tiếp cận kiến thức tiến bộ để chưng cất các mô hình biến áp nhỏ (70M) thành các mô hình Hyena. [6]

Giải mã suy đoán. Giải mã suy đoán [10, 11, 41, 67, 76] gần đây đã nổi lên như một phương pháp đầy hứa hẹn để đẩy nhanh quá trình suy luận của các mô hình ngôn ngữ lớn, đặc biệt là Transformers. Phương pháp này sử dụng một mô hình bản nháp nhỏ hơn để tạo ra các mã thông báo ứng viên theo suy đoán, sau đó mô hình mục tiêu lớn hơn sẽ xác minh. Chen et al. [11], Leviathan et al. [41] đã đề xuất một lược đồ lấy mẫu từ chối để cải thiện chất lượng suy luận, trong khi Spector và Re [67] đã sắp xếp các mã thông báo ứng viên thành một cấu trúc cây để cho phép xác minh hiệu quả hơn . Các công trình tiếp theo đã kiểm tra cả các mô hình bản nháp đã được đào tạo [5, 13, 48] và các mô hình bản nháp không cần đào tạo [24, 32, 78].

8 Kết luận

Chúng tôi xem xét vấn đề duy trì khả năng LLM trong khi tăng tốc độ giải mã thông qua sự kết hợp giữa chưng cất và giải mã suy đoán. Đầu tiên, chúng tôi chỉ ra rằng Transformer LLM có thể được sử dụng để khởi tạo hiệu quả mô hình RNN tuyến tính Mamba trong khi vẫn duy trì khả năng ban đầu. Sau đó, chúng tôi chỉ ra rằng thông qua sự kết hợp giữa chưng cất theo các hướng dẫn và sở thích có giám sát, chúng tôi có thể cải thiện khả năng của mô hình với tương đối ít tính toán. Cuối cùng, chúng tôi chỉ ra rằng mô hình Mamba có thể được tăng tốc đáng kể tại thời điểm suy luận thông qua việc sử dụng phương pháp giải mã suy đoán nhận biết phần cứng. Mô hình đầy đủ gần với độ chính xác trò chuyện LLM và được tăng tốc với giải mã suy đoán. Chúng tôi tin rằng những kết quả này cho thấy kiến thức về transformer có thể được chuyển giao hiệu quả sang các kiến trúc khác, mở ra tiềm năng tùy chỉnh hồ sơ suy luận của LLM ngoài việc tối ưu hóa sự chú ý.

Sự thừa nhận

Chúng tôi cảm ơn Together AI đã cung cấp tính toán cho một số thí nghiệm. Công trình này đã được hưởng lợi từ các cuộc thảo luận hữu ích với Albert Gu tại CMU, François Fleuret và Vincent Micheli tại Đại học Geneva, Albert Tseng và Wen-Ding Li tại Đại học Cornell. Chúng tôi cũng cảm ơn Xuhui Zhang từ Đại học Kỹ thuật Munich đã chỉ ra lỗi mặt nạ chú ý trong bản thảo đầu tiên.

Tài liệu tham khảo

[1] Arora, S., Eyuboglu, S., Timalsina, A., Johnson, I., Poli, M., Zou, J., Rudra, A., và Ré, C. (2023). Động vật học: Đo lường và cải thiện khả năng nhớ lại trong các mô hình ngôn ngữ hiệu quả. Bản in trước arXiv arXiv:2312.04927.

[2] Arora, S., Eyuboglu, S., Zhang, M., Timalsina, A., Alberti, S., Zinsley, D., Zou, J., Rudra, A., và Ré, C. (2024). Các mô hình ngôn ngữ chú ý tuyến tính đơn giản cân bằng sự đánh đổi giữa khả năng thu hồi và thông lượng. Bản in trước arXiv arXiv:2402.18668.

[3] BAAI (2024). Hướng dẫn vô cực. Bản in trước arXiv arXiv:2406.XXXX.

[4] Beck, M., Pöppel, K., Spanring, M., Auer, A., Prudnikova, O., Kopp, M., Klambauer, G., Brandstetter, J., và Hochreiter, S. (2024). xlstm: Bộ nhớ dài hạn mở rộng. Bản in trước arXiv arXiv:2405.04517.

[5] Bhendawade, N., Belousova, I., Fu, Q., Mason, H., Rastegari, M., và Najibi, M. (2024). suy đoán phát trực tuyến: Suy luận llm nhanh mà không cần mô hình phụ trợ. Bản in trước arXiv arXiv:2402.11131.

[6] Bick, A., Li, KY, Xing, EP, Kolter, JZ và Gu, A. (2024). Biến đổi thành ssms: Chắt lọc kiến thức bậc hai thành các mô hình dưới bậc hai. Bản in trước arXiv arXiv:2408.10189.

[7] Bisk, Y., Zellers, R., Gao, J., Choi, Y., et al. (2020). PIQA: Lý luận về ý nghĩa vật lý thông thường trong ngôn ngữ tự nhiên. Trong Biên bản Hội nghị AAAI về Trí tuệ nhân tạo, tập 34, trang 7432–7439.

[8] Botev, A., De, S., Smith, SL, Fernando, A., Muraru, G.-C., Haroun, R., Berrada, L., Pascanu, R., Sessa, PG, Dadashi, R., và những người khác. (2024). Recurrentgemma: Di chuyển các máy biến áp trong quá khứ cho các mô hình ngôn ngữ mở hiệu quả. bản in trước arXiv arXiv:2404.07839.

[9] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, JD, Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Các mô hình ngôn ngữ là những người học ít lần. Những tiến bộ trong hệ thống xử lý thông tin thần kinh, 33:1877–1901.

[10] Cai, T., Li, Y., Geng, Z., Peng, H., Lee, JD, Chen, D., và Dao, T. (2024). Medusa: Suy luận llm đơn giản khung tăng tốc với nhiều đầu giải mã. Bản in trước arXiv arXiv:2401.10774.

[11] Chen, C., Borgeaud, S., Irving, G., Lespiau, J.-B., Sifre, L., và Jumper, J. (2023a). Tăng tốc lớn Giải mã mô hình ngôn ngữ bằng phương pháp lấy mẫu suy đoán.

[12] Chen, J., Qadri, R., Wen, Y., Jain, N., Kirchenbauer, J., Zhou, T., và Goldstein, T. (2024). Genqa: Tạo ra hàng triệu hướng dẫn từ một số ít lời nhắc. Bản in trước arXiv arXiv:2406.10323.

[13] Chen, Z., Yang, X., Lin, J., Sun, C., Huang, J., và Chang, KC-C. (2023b). Soạn thảo suy đoán theo tầng để suy luận llm nhanh hơn. Bản in trước arXiv arXiv:2312.11462.

[14] Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., và Tafjord, O. (2018a). Bạn nghĩ mình đã giải quyết được Câu hỏi trả lời? Hãy thử ARC, Thử thách suy luận AI2. Bản in trước arXiv arXiv:1803.05457.

[15] Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., và Tafjord, O. (2018b). Bạn nghĩ mình đã giải quyết được câu hỏi trả lời? Hãy thử arc, thử thách suy luận ai2. Bản in trước arXiv arXiv:1803.05457.

[16] Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., et al. (2021). Đào tạo trình xác minh để giải các bài toán bằng lời. Bản in trước arXiv arXiv:2110.14168.

[17] Cui, G., Yuan, L., Ding, N., Yao, G., Zhu, W., Ni, Y., Xie, G., Liu, Z., và Sun, M. (2023). Siêu phản hồi: Tăng cường mô hình ngôn ngữ với phản hồi chất lượng cao. Bản in trước arXiv arXiv:2310.01377.

[18] Dao, T. và Gu, A. (2024). Máy biến áp là ssms: Các mô hình tổng quát và các thuật toán hiệu quả thông qua tính chất không gian trạng thái có cấu trúc. Bản in trước arXiv arXiv:2405.21060.

[19] De, S., Smith, SL, Fernando, A., Botev, A., Cristian-Muraru, G., Gu, A., Haroun, R., Berrada, L., Chen, Y., Srinivasan, S., et al. (2024). Griffin: Trộn các phép lặp tuyến tính có cổng với sự chú ý cục bộ để tạo ra các mô hình ngôn ngữ hiệu quả . Bản in trước arXiv arXiv:2402.19427.

[20] Ding, N., Chen, Y., Xu, B., Qin, Y., Hu, S., Liu, Z., Sun, M., và Zhou, B. (2023). Tăng cường các mô hình ngôn ngữ trò chuyện bằng cách mở rộng các cuộc trò chuyện hướng dẫn chất lượng cao. Trong Biên bản báo cáo Hội nghị năm 2023 về Phương pháp thực nghiệm trong Xử lý ngôn ngữ tự nhiên, trang 3029–3051.

[21] Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., et al. (2024). Mô hình đàn llama 3. Bản in trước arXiv arXiv:2407.21783.

[22] Fu, D., Arora, S., Grogan, J., Johnson, I., Eyuboglu, ES, Thomas, A., Spector, B., Poli, M., Rudra, A., và Ré, C. (2024a). Bộ trộn Monarch: Một kiến trúc đơn giản dựa trên gemm dưới bậc hai. Những tiến bộ trong Hệ thống xử lý thông tin thần kinh, 36.

[23] Fu, DY, Dao, T., Saab, KK, Thomas, AW, Rudra, A., và Re, C. (2022). Những con hà mã đói: Hướng tới mô hình hóa ngôn ngữ với các mô hình không gian trạng thái. Trong Hội nghị quốc tế lần thứ mười một về biểu diễn học tập.

[24] Fu, Y., Bailis, P., Stoica, I., và Zhang, H. (2024b). Phá vỡ sự phụ thuộc tuần tự của suy luận llm sử dụng giải mã lookahead. Bản in trước arXiv arXiv:2402.02057.

[25] Gao, L., Tow, J., Abbasi, B., Biderman, S., Black, S., DiPofi, A., Foster, C., Golding, L., Hsu, J., Le Noac'h, A., Li, H., McDonell, K., Muennighoff, N., Ociepa, C., Phang, J., Reynolds, L., Schoelkopf, H., Skowron, A., Sutawika, L., Tang, E., Thite, A., Wang, B., Wang, K., và Zou, A. (2023). Một khuôn khổ để đánh giá mô hình ngôn ngữ vài lần .

[26] Gu, A. và Dao, T. (2023). Mamba: Mô hình hóa chuỗi thời gian tuyến tính với không gian trạng thái chọn lọc. Bản in trước arXiv arXiv:2312.00752.

[27] Gu, A., Goel, K., Gupta, A., và Ré, C. (2022). Về tham số hóa và khởi tạo của đường chéo Mô hình không gian trạng thái. Những tiến bộ trong hệ thống xử lý thông tin thần kinh, 35:35971–35983.

[28] Gu, A., Goel, K., và Ré, C. (2021). Mô hình hóa hiệu quả các chuỗi dài với không gian trạng thái có cấu trúc. Bản in trước arXiv arXiv:2111.00396.

[29] Gu, A., Rozière, B., Leather, H., Solar-Lezama, A., Synnaeve, G., và Wang, SI (2024). Cruxeval: Điểm chuẩn cho lý luận, hiểu biết và thực thi mã. Bản in trước arXiv arXiv:2401.03065.

[30] Guo, M., Ainslie, J., Uthus, D., Ontanon, S., Ni, J., Sung, Y.-H., và Yang, Y. (2021). Longt5: Hiệu quả Bộ chuyển đổi văn bản sang văn bản cho các chuỗi dài. Bản in trước arXiv arXiv:2112.07916.

[31] Gupta, A., Gu, A., và Berant, J. (2022). Không gian trạng thái chéo có hiệu quả như không gian trạng thái có cấu trúc. Những tiến bộ trong hệ thống xử lý thông tin thần kinh, 35:22982–22994.

[32] He, Z., Zhong, Z., Cai, T., Lee, JD, và He, D. (2023). Phần còn lại: Giải mã suy đoán dựa trên truy xuất. Bản in trước arXiv arXiv:2311.08252.

[33] Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., và Steinhardt, J. (2020). Đo lường sự hiểu biết ngôn ngữ đa nhiệm vụ lớn. Trong Hội nghị quốc tế về biểu diễn học tập.

[34] Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., và Steinhardt, J. (2021). Đo lường khả năng hiểu ngôn ngữ đa nhiệm vụ lớn. Biên bản Hội nghị quốc tế về Biểu diễn học tập (ICLR).

[35] Hinton, G., Vinyals, O., và Dean, J. (2015). Chắt lọc kiến thức trong mạng nơ-ron. Bản in trước arXiv arXiv:1503.02531.

[36] Irie, K., Schlag, I., Csordás, R., và Schmidhuber, J. (2021). Vượt ra ngoài các bộ biến đổi tuyến tính với các lập trình viên trọng số nhanh tuần hoàn. Những tiến bộ trong hệ thống xử lý thông tin thần kinh, 34:7703–7717.

[37] Jiang, AQ, Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, DS, Casas, D. dl, Bressand, F., Lengyel, G., Lample, G., Saulnier, L., et al. (2023). Mistral 7b. bản in trước arXiv arXiv:2310.06825.

[38] Jin, Q., Dhingra, B., Liu, Z., Cohen, WW, và Lu, X. (2019). PubMedQA: Một tập dữ liệu cho Y sinh học Trả lời câu hỏi nghiên cứu. Bản in trước arXiv arXiv:1909.06146.

[39] Kim, Y. và Rush, AM (2016). Chưng cất kiến thức ở cấp độ trình tự. Trong Biên bản báo cáo năm 2016 Hội nghị về Phương pháp thực nghiệm trong Xử lý ngôn ngữ tự nhiên, trang 1317–1327.

[40] Lai, G., Xie, Q., Liu, H., Yang, Y., và Hovy, E. (2017). RACE: Hiểu biết về ReAding trên quy mô lớn Bộ dữ liệu từ các kỳ thi. Bản in trước arXiv arXiv:1704.04683.

[41] Leviathan, Y., Kalman, M., và Matias, Y. (2023). Suy luận nhanh từ máy biến áp thông qua giải mã suy đoán. Trong Biên bản báo cáo Hội nghị quốc tế lần thứ 40 về học máy, tập 202 của Biên bản báo cáo nghiên cứu học máy, trang 19274–19286. PMLR.

[42] Li, R., Allal, LB, Zi, Y., Muennighoff, N., Kocetkov, D., Mou, C., Marone, M., Akiki, C., Li, J., Chim, J., et al. (2023a). Starcoder: mong nguồn thông tin sẽ luôn ở bên bạn! bản in trước arXiv arXiv:2305.06161.

[43] Li, X., Zhang, T., Dubois, Y., Taori, R., Gulrajani, I., Guestrin, C., Liang, P., và Hashimoto, TB (2023b). Alpacaeval: Công cụ đánh giá tự động các mô hình tuân theo hướng dẫn. https://github.com/tatsu-lab/ alpaca_eval.

[44] Lieber, O., Lenz, B., Bata, H., Cohen, G., Osin, J., Dalmedigos, I., Safahi, E., Meirom, S., Belinkov, Y., Shalev-Shwartz, S., et al. (2024). Jamba: Một mô hình ngôn ngữ lai-mamba. bản in trước arXiv arXiv:2403.19887.

[45] Lin, BY (2024). ZeroEval: Một khuôn khổ thống nhất để đánh giá các mô hình ngôn ngữ.

[46] Lin, S., Hilton, J., và Evans, O. (2021). TruthfulQA: Đo lường cách các mô hình bắt chước sự dối trá của con người. Bản in trước arXiv arXiv:2109.07958.

[47] Lin, S., Hilton, J., và Evans, O. (2022). Truthfulqa: Đo lường cách các mô hình bắt chước sự dối trá của con người. Trong Biên bản Hội nghị thường niên lần thứ 60 của Hiệp hội Ngôn ngữ học tính toán (Tập 1: Bài báo dài), trang 3214–3252.

[48] Liu, X., Hu, L., Bailis, P., Stoica, I., Deng, Z., Cheung, A., và Zhang, H. (2023). Suy đoán trực tuyến giải mã. bản in trước arXiv arXiv:2310.07177.

[49] Massaroli, S., Poli, M., Fu, D., Kumbong, H., Parnichkun, R., Romero, D., Timalsina, A., McIntyre, Q., Chen, B., Rudra, A., et al. (2024). Nhà máy chưng cất linh cẩu cười: Trích xuất các phép lặp lại nhỏ gọn từ các phép tích chập. Những tiến bộ trong Hệ thống xử lý thông tin thần kinh, 36.

[50] Mehta, H., Gupta, A., Cutkosky, A., và Neyshabur, B. (2023). Mô hình hóa ngôn ngữ tầm xa thông qua Không gian nhà nước có cổng. Trong Hội nghị quốc tế lần thứ mười một về biểu diễn học tập.

[51] Meng, Y., Xia, M., và Chen, D. (2024). Simpo: Tối ưu hóa sở thích đơn giản với tham chiếu không có phần thưởng. bản in trước arXiv arXiv:2405.14734.

[52] Mercat, J., Vasiljevic, I., Keh, S., Arora, K., Dave, A., Gaidon, A., và Kollar, T. (2024). Tuyến tính hóa các mô hình ngôn ngữ lớn. Bản in trước arXiv arXiv:2405.06640.

[53] Merity, S., Xiong, C., Bradbury, J., và Socher, R. (2016). Các mô hình hỗn hợp lính canh chỉ điểm. Bản in trước arXiv arXiv:1609.07843.

[54] Mihaylov, T., Clark, P., Khot, T., và Sabharwal, A. (2018). Một bộ áo giáp có thể dẫn điện không? Bộ dữ liệu mới để trả lời câu hỏi Sách mở. Bản in trước arXiv arXiv:1809.02789.

[55] Peng, B., Alcaide, E., Anthony, Q., Albalak, A., Arcadinho, S., Cao, H., Cheng, X., Chung, M., Grella, M., GV, KK, et al. (2023a). Rwkv: Tái phát minh rnn cho kỷ nguyên máy biến áp. bản in trước arXiv arXiv:2305.13048.

[56] Peng, B., Quesnelle, J., Fan, H., và Shippole, E. (2023b). Yarn: Mở rộng cửa sổ ngữ cảnh hiệu quả của mô hình ngôn ngữ lớn. bản in trước arXiv arXiv:2309.00071.

[57] Poli, M., Massaroli, S., Nguyen, E., Fu, DY, Dao, T., Baccus, S., Bengio, Y., Ermon, S., và Ré, C. (2023). Hệ thống phân cấp linh cẩu: Hướng tới các mô hình ngôn ngữ tích chập lớn hơn. Trong Hội nghị quốc tế về học máy, trang 28043–28078. PMLR.

[58] Rafailov, R., Sharma, A., Mitchell, E., Manning, CD, Ermon, S., và Finn, C. (2024). Tối ưu hóa sở thích trực tiếp: Mô hình ngôn ngữ của bạn thực chất là một mô hình phần thưởng. Những tiến bộ trong Hệ thống xử lý thông tin thần kinh, 36.

[59] Ralambomihanta, TR, Mohammadzadeh, S., Islam, MSN, Jabbour, W., và Liang, L. (2024). Linh cẩu ăn xác thối: Chắt lọc các bộ biến đổi thành các mô hình tích chập dài. Bản in trước của arXiv arXiv:2401.17574.

[60] Ren, L., Liu, Y., Lu, Y., Shen, Y., Liang, C., và Chen, W. (2024). Samba: Các mô hình không gian trạng thái lai đơn giản để mô hình hóa ngôn ngữ ngữ cảnh không giới hạn hiệu quả. Bản in trước arXiv arXiv:2406.07522.

[61] Roziere, B., Gehring, J., Gloeckle, F., Sootla, S., Gat, I., Tan, XE, Adi, Y., Liu, J., Remez, T., Rapin, J., et al. (2023). Code llama: Mô hình nền tảng mở cho code. Bản in trước arXiv arXiv:2308.12950.

[62] Sakaguchi, K., Bras, RL, Bhagavatula, C., và Choi, Y. (2021). Winogrande: Một thách thức về lược đồ winograd đối đầu ở quy mô lớn. Truyền thông của ACM, 64(9):99–106.

[63] Schlag, I., Irie, K., và Schmidhuber, J. (2021). Máy biến áp tuyến tính là bộ lập trình trọng số nhanh một cách bí mật. Trong Hội nghị quốc tế về máy học, trang 9355–9366. PMLR.

[64] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., và Klimov, O. (2017). Tối ưu hóa chính sách gần thuật toán. bản in trước arXiv arXiv:1707.06347.

[65] Shaham, U., Segal, E., Ivgi, M., Efrat, A., Yoran, O., Haviv, A., Gupta, A., Xiong, W., Geva, M., Berant, J., et al. (2022). Cuộn giấy: So sánh chuẩn hóa trên các chuỗi ngôn ngữ dài. Bản in trước arXiv arXiv:2201.03533.

[66] Shleifer, S. và Rush, AM (2020). Chưng cất tóm tắt được đào tạo trước. CoRR, abs/2010.13002.

[67] Spector, B. và Re, C. (2023). Tăng tốc suy luận llm với giải mã suy đoán theo giai đoạn. Bản in trước arXiv arXiv:2308.04623.

[68] Sun, Y., Dong, L., Huang, S., Ma, S., Xia, Y., Xue, J., Wang, J., và Wei, F. (2023). Mạng lưới lưu giữ: A người kế nhiệm bộ chuyển đổi cho các mô hình ngôn ngữ lớn. bản in trước arXiv arXiv:2307.08621.

[69] Tay, Y., Dehghani, M., Abnar, S., Shen, Y., Bahri, D., Pham, P., Rao, J., Yang, L., Ruder, S., và Metzler, D. (2020). Đấu trường tầm xa: Điểm chuẩn cho máy biến áp hiệu quả. Trong Hội nghị quốc tế về Biểu diễn học tập.

[70] Teknium (2023). Openhermes 2.5: Một tập dữ liệu mở về dữ liệu tổng hợp dành cho trợ lý llm tổng quát.

[71] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., và những người khác. (2023). Llama: Các mô hình ngôn ngữ nền tảng mở và hiệu quả. bản in trước arXiv arXiv:2302.13971.

[72] Tunstall, L., Beeching, E., Lambert, N., Rajani, N., Rasul, K., Belkada, Y., Huang, S., von Werra, L., Fourrier, C., Habib, N., et al. (2023). Zephyr: Chưng cất trực tiếp sự liên kết lm. bản in trước arXiv arXiv:2310.16944.

[73] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, AN, Kaiser, Ł., và Polosukhin, I. (2017). Sự chú ý là tất cả những gì bạn cần. Những tiến bộ trong hệ thống xử lý thông tin thần kinh, 30.

[74] Waleffe, R., Byeon, W., Riach, D., Norick, B., Korthikanti, V., Dao, T., Gu, A., Hatamizadeh, A., Singh, S., Narayanan, D., et al. (2024). Một nghiên cứu thực nghiệm về các mô hình ngôn ngữ dựa trên mamba. bản in trước arXiv arXiv:2406.07887.

[75] Wang, J., Yan, JN, Gu, A., và Rush, AM (2022). Đào tạo trước mà không cần chú ý. Bản in trước arXiv arXiv:2212.10544.

[76] Xia, H., Ge, T., Wang, P., Chen, S.-Q., Wei, F., và Sui, Z. (2023). Giải mã suy đoán: Khai thác thực thi suy đoán để tăng tốc tạo Seq2seq. Trong Phát hiện của Hiệp hội Ngôn ngữ học tính toán: EMNLP 2023, trang 3909–3925, Singapore. Hiệp hội Ngôn ngữ học tính toán.

[77] Yang, J., Jimenez, CE, Wettig, A., Lieret, K., Yao, S., Narasimhan, K., và Press, O. (2024a). Đại lý Swe: Giao diện máy tính của tác nhân cho phép xây dựng mô hình ngôn ngữ kỹ thuật phần mềm.

[78] Yang, N., Ge, T., Wang, L., Jiao, B., Jiang, D., Yang, L., Majumder, R., và Wei, F. (2023a). Suy luận có tham chiếu: Tăng tốc không mất dữ liệu của các mô hình ngôn ngữ lớn. Bản in trước arXiv arXiv:2304.04487.

[79] Yang, S., Wang, B., Shen, Y., Panda, R., và Kim, Y. (2023b). Máy biến áp chú ý tuyến tính có cổng với đào tạo hiệu quả về phần cứng. Bản in trước arXiv arXiv:2312.06635.

[80] Yang, S., Wang, B., Zhang, Y., Shen, Y., và Kim, Y. (2024b). Song song hóa các máy biến áp tuyến tính với quy tắc delta trên chiều dài chuỗi. bản in trước arXiv arXiv:2406.06484.

[81] Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., và Cao, Y. (2022). Phản ứng: Tương tác lý luận và hành động trong các mô hình ngôn ngữ. bản in trước arXiv arXiv:2210.03629.

[82] Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., và Choi, Y. (2019). Hellaswag: Một cỗ máy thực sự có thể hoàn thành câu của bạn? bản in trước arXiv arXiv:1905.07830.

[83] Zhang, M., Bhatia, K., Kumbong, H., và Ré, C. (2024). Nhím và nhím: Biểu cảm sự chú ý tuyến tính với sự mô phỏng softmax. bản in trước arXiv arXiv:2402.04347.

[84] Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, EP, Zhang, H., Gonzalez, JE và Stoica, I. (2023). Đánh giá llm-as-a-thẩm phán với đấu trường mt-bench và chatbot.