गहन संवाद

मुझे □□□□□□□□□-□3 तकनीकी रिपोर्ट को पढ़ने में काफी समय लगा है, और मुझे इस मॉडल की पैमाने की वजह से बहुत प्रभावित हुआ है। 671 अरब पैरामीटर, लेकिन केवल 37 अरब टोकन प्रति सक्रिय? यह एक बहुत बड़ा □□□ आर्किटेक्चर है। यह कैसे काम करता है?

□: बिल्कुल, यह एक बड़ा काम है! □□□□□□□□□-□3 □□□□□□□□-□□-□□□□□□□□ (□□□) फ्रेमवर्क पर बनाया गया है, जो प्रत्येक टोकन के लिए केवल एक उपसमूह पैरामीटर सक्रिय करने की अनुमति देता है। विशेष रूप से, यह 256 रूटेड एक्सपर्ट्स का उपयोग करता है, लेकिन प्रत्येक टोकन के लिए केवल 8 सक्रिय होते हैं। यह घन मॉडलों की तुलना में बहुत अधिक दक्ष है, जहां प्रत्येक टोकन के लिए सभी पैरामीटर सक्रिय होते हैं।

□: यह समझ में आता है। लेकिन यह कैसे फैसला करता है कि कौन से एक्सपर्ट सक्रिय होंगे? क्या यह बस रैंडम है, या कोई तरह का रूटिंग मैकेनिज्म है?

□: एक अच्छा सवाल है! रूटिंग टोकन-से-एक्सपर्ट एफिनिटी स्कोर पर आधारित है। प्रत्येक टोकन को प्रत्येक एक्सपर्ट के लिए एक स्कोर दिया जाता है, और सबसे ऊंचे स्कोर वाले टॉप-□ एक्सपर्ट सक्रिय होते हैं। □□□□□□□□□-□3 एक सिग्मॉइड फंक्शन का उपयोग करके इन स्कोरों को कंप्यूट करता है, जो एक्सपर्ट्स के बीच लोड को संतुलित करने में मदद करता है।

□: अह, तो यह सीखने के दौरान सीखा जाता है। लेकिन क्या यह एक्सपर्ट उपयोग में असंतुलन को नहीं लाता? मुझे लगता है कि यह □□□ मॉडलों में एक आम समस्या है।

□: बिल्कुल! असंतुलित एक्सपर्ट उपयोग एक समस्या हो सकती है, लेकिन □□□□□□□□□-□3 एक ऑक्सिलरी-लॉस-फ्री स्ट्रैटेजी का उपयोग करता है इसे संभालने के लिए। इसके बजाय एक अलग लॉस टर्म को लोड संतुलन को प्रोत्साहित करने के लिए जोड़ने के बजाय, यह प्रत्येक एक्सपर्ट के लिए एक बायस टर्म को डायनामिक रूप से अद्यतन करता है। अगर एक एक्सपर्ट ओवरलोड है, तो उसका बायस कम किया जाता है, और अगर वह अंडरलोड है, तो बायस बढ़ाया जाता है। यह लोड को संतुलित करता है बिना मॉडल प्रदर्शन को खराब करने के।

□: यह चतुर है। तो, कोई ऑक्सिलरी लॉस का मतलब है कम हस्तक्षेप मुख्य प्रशिक्षण उद्देश्य के साथ। लेकिन यह पारंपरिक □□□ मॉडलों के साथ कैसे तुलना करता है जो ऑक्सिलरी लॉस का उपयोग करते हैं?

□: बिल्कुल। पारंपरिक □□□ मॉडल अक्सर लोड संतुलन को प्रोत्साहित करने के लिए ऑक्सिलरी लॉस का उपयोग करते हैं, लेकिन ये लॉस कभी-कभी प्रदर्शन को नुकसान पहुंचा सकते हैं। □□□□□□□□□-□3 के ऑक्सिलरी-लॉस-फ्री एप्रोच इस ट्रेड-ऑफ को टाल देता है। वास्तव में, एब्लेशन अध्ययनों से पता चलता है कि यह मॉडलों से बेहतर प्रदर्शन करता है जो ऑक्सिलरी लॉस पर निर्भर करते हैं, खासकर कोडिंग और गणित जैसे टास्क्स पर।

□: रोचक। कोडिंग और गणित के बारे में बात करते हुए, मैंने देखा कि □□□□□□□□□-□3 □□□□□□□□□□ और □□□□ जैसी बेंचमार्क पर बेहतरीन प्रदर्शन करता है। वहाँ की रसोई क्या है?

□: इसका एक बड़ा हिस्सा है मल्टी-टोकन प्रेडिक्शन (□□□) उद्देश्य। इसके बजाय केवल अगले टोकन को प्रेडिक्ट करने के बजाय, □□□□□□□□□-□3 प्रत्येक स्थिति पर कई भविष्यवाणी टोकन प्रेडिक्ट करता है। यह ट्रेनिंग सिग्नल को घन करता है और मॉडल को आगे सोचने में मदद करता है, जो कोडिंग और गणित जैसे टास्क्स के लिए विशेष रूप से उपयोगी है।

□: इंतजार, तो यह एक साथ कई टोकन का प्रेडिक्शन कर रहा है? यह इन्फरेंस के दौरान कैसे काम करता है? क्या यह अभी भी □□□ का उपयोग करता है, या यह केवल ट्रेनिंग के लिए है?

□: इन्फरेंस के दौरान, □□□ मॉड्यूल को छोड़ दिया जा सकता है, और मॉडल एक सामान्य ऑटोरेग्रेसिव मॉडल की तरह व्यवहार करता है। लेकिन यहाँ की चालाकी है: □□□ मॉड्यूल को भी स्पेकुलेटिव डिकोडिंग के लिए पुनः उपयोग किया जा सकता है, जो कई टोकन को एक साथ भविष्यवाणी करके और फिर उन्हें सत्यापित करके जनरेशन को तेज करता है।

□: यह एक चतुर ट्रिक है। तो, यह ट्रेनिंग के दौरान □□□ के फायदे को प्राप्त करता है और फिर इसे इन्फरेंस को तेज करने के लिए उपयोग करता है। लेकिन ध्यान केंद्रित करने की मैकेनिज्म के बारे में? मैंने कुछ □□□□□-□□□□ □□□□□□ □□□□□□□□□□ (□□□) के बारे में देखा। यह कैसे फिट होता है?

□: □□□ एक और महत्वपूर्ण नवाचार है। यह □□□-□□□□□ (□□) कैश को कम करता है, जो पूर्ण ध्यान के कुंजी और मानों को स्टोर करने के बजाय, कम-रैंक जॉइंट कम्प्रेशन का उपयोग करके उन्हें प्रतिनिधित्व करता है। यह इन्फरेंस के दौरान □□ कैश आकार को काफी कम करता है जबकि प्रदर्शन को स्टैंडर्ड मल्टी-हेड

एटेंशन के समान बनाए रखता है।

□: यह दक्षता में एक बड़ा जीत है। लेकिन कम्प्रेशन में कुछ जानकारी का नुकसान नहीं होता? यह प्रदर्शन कैसे बनाए रखता है?

□: एक अच्छा सवाल है। कम्प्रेशन को सबसे महत्वपूर्ण जानकारी को संरक्षित करने के लिए डिज़ाइन किया गया है, जो कुंजी और मानों के महत्वपूर्ण विशेषताओं को पकड़ने वाले लेटेंट वेक्टरों पर ध्यान केंद्रित करता है। मॉडल भी □□□□□□ □□□□□□□□□□ □□□□□□□□□□ (□□□□) का उपयोग करता है, जो कम्प्रेशन से होने वाले किसी भी नुकसान को कम करने में मदद करता है।

□: समझ में आया। तो, □□□ दक्षता को बढ़ाने के लिए है बिना बहुत अधिक प्रदर्शन को बलि देना। लेकिन प्रशिक्षण के बारे में? इस तरह के मॉडल को प्रशिक्षित करना बहुत महंगा होगा। □□□□□□□□-□3 कैसे लागत को कम रखता है?

□: प्रशिक्षण दक्षता एक बड़ा ध्यान केंद्र है। □□□□□□□□-□3 एक □□8 मिक्स्ड प्रिसिजन फ्रेमवर्क का उपयोग करता है, जो मेमोरी उपयोग को कम करता है और गणना को तेज करता है। यह भी एक □□□□□□□□ एल्गोरिथ□ का उपयोग करता है, जो पाइपलाइन पैरललिज्म के लिए, जो पाइपलाइन बबल्स को न्यूनतम करता है और गणना को संचार के साथ ओवरलैप करता है। इन ऑप्टिमाइजेशन के कारण मॉडल को केवल 2.788 मिलियन □800 □□□ घंटों में 14.8 ट्रिलियन टोकन पर प्रशिक्षित किया जा सकता है।

□: यह प्रभावशाली है। लेकिन □□8 प्रशिक्षण मुश्किल हो सकता है—वे प्रिसिजन समस्याओं को कैसे संभालते हैं? मुझे लगता है कि कम-प्रिसिजन प्रशिक्षण अस्थिरता ला सकता है।

□: आप बिल्कुल सही हैं। □□8 प्रशिक्षण चुनौतीपूर्ण है क्योंकि इसके सीमित डायनामिक रेंज के कारण। □□□□□□□□-□3 इसको हल करने के लिए फाइन-ग्रेनेड क्वांटाइजेशन का उपयोग करता है, जहां एक्टिवेशन और वेट्स को छोटे टाइल या ब्लॉक में समूहित किया जाता है और स्वतंत्र रूप से स्केल किया जाता है। यह आउटलियर्स को संभालने में मदद करता है और प्रशिक्षण को स्थिर रखता है। वे भी महत्वपूर्ण ऑपरेशनों के लिए उच्च-प्रिसिजन अक्यूमुलेशन का उपयोग करते हैं, जैसे मैट्रिक्स गणना, ताकि सटीकता बनाए रखें।

□: यह समझ में आता है। तो, यह दक्षता और प्रिसिजन के बीच एक संतुलन है। लेकिन डेटा के बारे में? 14.8 ट्रिलियन टोकन एक बड़ा डेटासेट है। यह किस तरह का डेटा है?

□: डेटासेट विविध और उच्च गुणवत्ता वाला है, जिसमें अंग्रेजी और चीनी टेक्स्ट पर ध्यान केंद्रित किया गया है। इसमें काफी गणित और प्रोग्रामिंग डेटा भी शामिल है, जो मॉडल को इन डोमेन में बेहतर बनाता है। डेटा पाइपलाइन को कम करने के लिए ऑप्टिमाइज किया जाता है जबकि विविधता बनाए रखता है, और वे डॉक्यूमेंट पैकिंग जैसे तकनीकों का उपयोग करते हैं ताकि डेटा की एकता बनाए रखें।

□: यह कोडिंग और गणित टास्क्स पर मजबूत प्रदर्शन को समझाता है। लेकिन बहुभाषी प्रदर्शन के बारे में? क्या यह अन्य भाषाओं को अच्छी तरह से संभालता है?

□: हाँ, □□□□□□□□-□3 एक बहुभाषी कॉरपस पर प्रशिक्षित है, और यह □□□□□ जैसी बेंचमार्क पर अच्छी तरह से प्रदर्शन करता है, जिसमें अंग्रेजी के बाहर के टास्क्स शामिल हैं। यह चीनी में विशेष रूप से मजबूत है, और चीनी बेंचमार्क जैसे □-□□□□ और □□□□□ पर □□□□2.5 जैसी मॉडलों से बेहतर प्रदर्शन करता है।

□: यह प्रभावशाली है। लेकिन लंबे-संदर्भ टास्क्स के बारे में? मैंने देखा कि यह 128□ टोकन तक का समर्थन करता है। यह कैसे लंबे इनपुट्स को संभालता है?

□: □□□□□□□□-□3 अपने संदर्भ लंबाई को दो चरणों में बढ़ाता है: पहले 32□ टोकन तक और फिर 128□ टोकन तक □□□□ तकनीक का उपयोग करके। यह लंबे-संदर्भ टास्क्स जैसे डॉक्यूमेंट सारांश और रिट्रीवल को प्रभावी रूप से संभालने में मदद करता है। यह ‘□□□□□□ □□ □ □□□□□□□□’टेस्ट पर भी अच्छी तरह से प्रदर्शन करता है, जो लंबे-संदर्भ समझ को मूल्यांकन करता है।

□: यह पिछले मॉडलों से एक बड़ा सुधार है। लेकिन डिप्लॉयमेंट के बारे में? वे इस तरह के बड़े मॉडल के लिए इन्फरेंस कैसे संभालते हैं?

□: इन्फरेंस एक □800 क्लस्टर पर संभाला जाता है, जहां □□□ □□□□□□ और □□□□□□□□□□ के साथ इंटरकनेक्टेड हैं। डिप्लॉयमेंट स्ट्रैटेजी प्री-फिलिंग और डिकोडिंग चरणों को अलग करता है ताकि उच्च थ्रूपुट और कम लेटेंसी सुनिश्चित किया जा सके। वे इन्फरेंस के दौरान लोड को संतुलित करने के लिए अतिरिक्त एक्सपर्ट्स का उपयोग भी करते हैं, जो दक्षता बनाए रखने में मदद करता है।

□: यह काफी ऑप्टिमाइजेशन हैं। लेकिन सीमाएं? निश्चित रूप से, इस तरह के बड़े मॉडल में कुछ ट्रेड-ऑफ होंगे।

□: एक सीमा डिप्लॉयमेंट यूनिट आकार है। □□□□□□□□□-□3 को दक्ष इन्फरेंस के लिए एक बड़ा क्लस्टर की आवश्यकता होती है, जो छोटे टीमों के लिए एक चुनौती हो सकती है। जनरेशन स्पीड में सुधार करने के लिए भी जगह है, हालांकि □□□ के साथ स्पेकुलेटिव डिकोडिंग मदद करता है।

□: ठीक है। लेकिन कुल मिलाकर, यह एक बड़ा कदम आगे लगता है। □□□□□□□□□-□3 के लिए अगला क्या है? वे किसी भी भविष्य के दिशाओं की ओर देख रहे हैं?

□: वे कई क्षेत्रों में काम कर रहे हैं, जैसे कि आर्किटेक्चर को संशोधित करने के लिए अनंत संदर्भ लंबाई का समर्थन करने, अतिरिक्त ट्रेनिंग सिग्नल स्रोतों का पता लगाने, और मॉडल के रीजनिंग क्षमता को बढ़ाने। वे भी मॉडल प्रदर्शन को बेहतर मूल्यांकन करने के लिए अधिक व्यापक मूल्यांकन विधियों पर काम कर रहे हैं।

□: लगता है वे जल्दी नहीं रुक रहे हैं। मुझे सब कुछ समझाने के लिए धन्यवाद—□□□□□□□□□-□3 निश्चित रूप से ओपन-सोर्स □□□ स्पेस में एक गेम-चेंजर है।

□: बिल्कुल! यह देखने में रोमांचक है कि ओपन-सोर्स मॉडल कितनी दूर पहुंच गए हैं। □□□□□□□□□-□3 सीमाओं को तोड़ रहा है, और मैं देखना चाहता हूँ कि वे अगला क्या करते हैं।

□: आपने कहा कि □□□□□□□□□-□3 □□8 मिक्स्ड प्रिसिजन ट्रेनिंग का उपयोग करता है। मुझे यह जानने में रुचि है—यह □□16 या □□16 के साथ कैसे तुलना करता है? क्या □□8 इस तरह के बड़े मॉडल को प्रशिक्षित करने के लिए पर्याप्त स्थिर है?

□: एक अच्छा सवाल है। □□8 वास्तव में सीमित डायनामिक रेंज के कारण चुनौतीपूर्ण है, लेकिन □□□□□□□□□-□3 फाइन-ग्रेनेड क्वांटाइजेशन स्ट्रैटेजी का उपयोग करता है इसे कम करने के लिए। उदाहरण के लिए, एक्टिवेशन को 1□128 टाइल में समूहित किया जाता है, और वेट्स को 128□128 ब्लॉक में समूहित किया जाता है। प्रत्येक समूह स्वतंत्र रूप से स्केल किया जाता है, जो आउटलियर्स को संभालने में मदद करता है और प्रशिक्षण को स्थिर रखता है।

□: रोचक। तो, यह बस एक □□8 क्वांटाइजेशन नहीं है—यह अधिक सूक्ष्म है। लेकिन क्या यह सभी इन समूहों और स्केलिंग कारकों को प्रबंधित करने के लिए अतिरिक्त ओवरहेड लाता है?

□: यह करता है, लेकिन लाभों के मुकाबले ओवरहेड बहुत कम है। मुख्य बात यह है कि □□8 मेमोरी उपयोग को कम करता है और गणना को तेज करता है, जो इस तरह के बड़े मॉडल को प्रशिक्षित करने के लिए महत्वपूर्ण है। वे भी महत्वपूर्ण ऑपरेशनों के लिए उच्च-प्रिसिजन अक्यूमुलेशन का उपयोग करते हैं, जैसे मैट्रिक्स गणना, ताकि न्यूमेरिकल स्थिरता सुनिश्चित की जा सके।

□: समझ में आया। तो, यह दक्षता और प्रिसिजन के बीच एक संतुलन है, लेकिन उन्होंने एक अच्छा संतुलन बनाया है। लेकिन □□□□□□□□ एल्गोरिथ□ के बारे में? यह कैसे काम करता है?

□: □□□□□□□□ पाइपलाइन पैरललिज्म में पाइपलाइन बबल्स को न्यूनतम करने के लिए डिज़ाइन किया गया है। यह प्रत्येक चंक ऑफ वर्क को चार घटकों में विभाजित करता है: ध्यान, ऑल-टू-ऑल डिस्पैच, □□□, और ऑल-टू-ऑल कॉम्बाइन। वापसी पास में, यह गणना को 'वापसी के लिए इनपुट' और 'वापसी के लिए वेट्स' में और विभाजित करता है, जो अधिक दक्ष ओवरलैप की अनुमति देता है।

□: यह जटिल लगता है, लेकिन समझ में आता है। तो, यह संचार ओवरहेड को गणना के साथ ओवरलैप करके छिपा रहा है। यह 1□1□ या □□□□ □□□□□□ जैसी अन्य पाइपलाइन पैरललिज्म विधियों के साथ कैसे तुलना करता है?

□: □□□□□□□□ में 1□1□ और □□□□ □□□□□□ की तुलना में कम पाइपलाइन बबल्स होते हैं। यह भी द्विधा शेड्यूलिंग की अनुमति देता है, जहां माइक्रो-बैच्स को पाइपलाइन के दोनों सिरों से फीड किया जाता है। यह खाली समय को और कम करता है और कुल दक्षता को बढ़ाता है। वास्तव में, □□□□□□□□ लगभग शून्य ऑल-टू-ऑल संचार ओवरहेड प्राप्त करता है, जो □□□ मॉडलों को बढ़ाने के लिए महत्वपूर्ण है।

□: यह प्रभावशाली है। लेकिन मेमोरी उपयोग के बारे में? क्या □□□□□□□□ अन्य विधियों की तुलना में अधिक मेमोरी का उपयोग करता है?

□: यह करता है, क्योंकि यह मॉडल पैरामीटर के दो कॉपी रखता है, लेकिन बढ़ोतरी प्रबंधनीय है। मेमोरी फुटप्रिंट को □□□□□□□ और □□□ अप-प्रोजेक्शंस के पुनः गणना जैसे तकनीकों के माध्यम से ऑप्टिमाइज किया जाता है, जो मध्यवर्ती एक्टिवेशन को स्टोर करने की आवश्यकता को हटा देता है।

□: अह, तो वे कुछ मेमोरी के लिए बेहतर दक्षता के लिए ट्रेड कर रहे हैं। यह एक अच्छा ट्रेड-ऑफ लगता है। मेमोरी के बारे में बात करते हुए, वे 128□ टोकन के लिए इस तरह के लंबे संदर्भ लंबाई के लिए □□ कैश को कैसे संभालते हैं?

□: यही है जहाँ □□□ सचमुच चमकता है। □□ कैश को कम्प्रेस करके, वे इसका आकार काफी कम करते हैं। पूर्ण ध्यान के कुंजी और मानों को स्टोर करने के बजाय, वे कम्प्रेस्ड लेटेंट वेक्टरों को स्टोर करते हैं, जो बहुत छोटे होते हैं। यह मॉडल को लंबे संदर्भों को संभालने में मदद करता है बिना मेमोरी बॉटलनेक्स में फंसने के।

□: यह एक चतुर हल है। लेकिन ध्यान की गुणवत्ता के बारे में? क्या कम्प्रेशन मॉडल को सही टोकन पर ध्यान देने की क्षमता को प्रभावित करता है?

□: कम्प्रेशन को सबसे महत्वपूर्ण जानकारी को संरक्षित करने के लिए डिज़ाइन किया गया है, इसलिए ध्यान की गुणवत्ता पर प्रभाव न्यूनतम है। वे भी □□□□ (□□□□□□ □□□□□□□□□□ □□□□□□□□□□) का उपयोग करते हैं, जो कम्प्रेशन से होने वाले किसी भी नुकसान को कम करने में मदद करता है।

□: समझ में आया। तो, □□□ एक जीत-जीत है—यह मेमोरी उपयोग को कम करता है बिना बहुत अधिक प्रदर्शन को बलि देना। लेकिन प्रशिक्षण डेटा के बारे में? आपने कहा कि यह 14.8 ट्रिलियन टोकन है। वे इस तरह के बड़े डेटासेट की गुणवत्ता और विविधता को कैसे सुनिश्चित करते हैं?

□: डेटासेट को उच्च गुणवत्ता और विविध टोकन शामिल करने के लिए सावधानी से संकलित किया जाता है। वे डेटा पाइपलाइन को कम करने के लिए ऑप्टिमाइज करते हैं जबकि विविधता बनाए रखते हैं, और वे डॉक्यूमेंट पैकिंग जैसे तकनीकों का उपयोग करते हैं ताकि डेटा की एकता सुनिश्चित की जा सके। कॉरपस में अंग्रेजी और चीनी टेक्स्ट का मिश्रण शामिल है, जिसमें गणित और प्रोग्रामिंग नमूने पर विशेष ध्यान दिया जाता है।

□: यह कोडिंग और गणित टास्क्स पर मजबूत प्रदर्शन को समझाता है। लेकिन बहुभाषी टास्क्स के बारे में? क्या यह अन्य भाषाओं को अच्छी तरह से संभालता है?

□: हाँ, □□□□□□□□-□3 एक बहुभाषी कॉरपस पर प्रशिक्षित है, और यह □□□□□ जैसी बेंचमार्क पर अच्छी तरह से प्रदर्शन करता है, जिसमें अंग्रेजी के बाहर के टास्क्स शामिल हैं। यह चीनी में विशेष रूप से मजबूत है, और चीनी बेंचमार्क जैसे □-□□□□ और □□□□□ पर □□□□2.5 जैसी मॉडलों से बेहतर प्रदर्शन करता है।

□: यह प्रभावशाली है। लेकिन लंबे-संदर्भ टास्क्स के बारे में? मैंने देखा कि यह 128□ टोकन तक का समर्थन करता है। यह कैसे लंबे इनपुट्स को संभालता है?

□: □□□□□□□□-□3 अपने संदर्भ लंबाई को दो चरणों में बढ़ाता है: पहले 32□ टोकन तक और फिर 128□ टोकन तक □□□□ तकनीक का उपयोग करके। यह लंबे-संदर्भ टास्क्स जैसे डॉक्यूमेंट सारांश और रिट्रीवल को प्रभावी रूप से संभालने में मदद करता है। यह '□□□□□□ □□ □ □□□□□□□□'टेस्ट पर भी अच्छी तरह से प्रदर्शन करता है, जो लंबे-संदर्भ समझ को मूल्यांकन करता है।

□: यह पिछले मॉडलों से एक बड़ा सुधार है। लेकिन डिप्लॉयमेंट के बारे में? वे इस तरह के बड़े मॉडल के लिए इन्फरेंस कैसे संभालते हैं?

□: इन्फरेंस एक □800 क्लस्टर पर संभाला जाता है, जहां □□□ □□□□□□ और □□□□□□□□□□ के साथ इंटरकनेक्टेड हैं। डिप्लॉयमेंट स्ट्रैटेजी प्री-फिलिंग और डिकोडिंग चरणों को अलग करता है ताकि उच्च थ्रूपुट और कम लेटेंसी सुनिश्चित किया जा सके। वे इन्फरेंस के दौरान लोड को संतुलित करने के लिए अतिरिक्त एक्सपर्ट्स का उपयोग भी करते हैं, जो दक्षता बनाए रखने में मदद करता है।

□: यह काफी ऑप्टिमाइजेशन हैं। लेकिन सीमाएं? निश्चित रूप से, इस तरह के बड़े मॉडल में कुछ ट्रेड-ऑफ होंगे।

□: एक सीमा डिप्लॉयमेंट यूनिट आकार है। □□□□□□□□-□3 को दक्ष इन्फरेंस के लिए एक बड़ा क्लस्टर की आवश्यकता होती है, जो छोटे टीमों के लिए एक चुनौती हो सकती है। जनरेशन स्पीड में सुधार करने के लिए भी जगह है, हालांकि □□□ के साथ स्पेकुलेटिव डिकोडिंग मदद करता है।

□: ठीक है। लेकिन कुल मिलाकर, यह एक बड़ा कदम आगे लगता है। □□□□□□□□-□3 के लिए अगला क्या है? वे किसी भी भविष्य के दिशाओं की ओर देख रहे हैं?

□: वे कई क्षेत्रों में काम कर रहे हैं, जैसे कि आर्किटेक्चर को संशोधित करने के लिए अनंत संदर्भ लंबाई का समर्थन करने, अतिरिक्त ट्रेनिंग सिग्नल स्रोतों का पता लगाने, और मॉडल के रीजनिंग क्षमता को बढ़ाने। वे भी मॉडल प्रदर्शन को बेहतर मूल्यांकन करने के लिए अधिक व्यापक मूल्यांकन विधियों पर काम कर रहे हैं।

□: लगता है वे जल्दी नहीं रुक रहे हैं। मुझे सब कुछ समझाने के लिए धन्यवाद—□□□□□□□□-□3 निश्चित रूप से ओपन-सोर्स □□□ स्पेस में एक गेम-चेंजर है।

□: बिल्कुल! यह देखने में रोमांचक है कि ओपन-सोर्स मॉडल कितनी दूर पहुंच गए हैं। □□□□□□□□-□3 सीमाओं को तोड़ रहा है, और मैं देखना चाहता हूँ कि वे अगला क्या करते हैं।