Deep Dive into LLMs like ChatGPT
深度剖析类似 ChatGPT 的大语言模型
视频链接：https://www.youtube.com/watch?v=7xTGNNLPyMI
ai总结：https://glasp.co/youtube/7xTGNNLPyMI?ref=youtube-summary

(00:01) hi everyone so I've wanted to make this video for a while it is a comprehensive but General audience introduction to large language models like Chachi PT and what I'm hoping to achieve in this video is to give you kind of mental models for thinking through what it is that this tool is it is obviously magical and amazing in some respects it's uh really good at some things not very good at other things and there's also a lot of sharp edges to be aware of so what is behind this text box you can put anything in there and press enter but uh
大家好，我一直想做这个视频有一段时间了。这是一个面向普通大众的、全面的关于像 ChatGPT 这样的大语言模型的介绍。我希望在这个视频中达到的目的是，给大家一些思维模型，让你们能够思考清楚这个工具到底是什么。在某些方面，它显然非常神奇和令人惊叹，它在某些事情上非常擅长，而在其他事情上则不是很擅长，并且还有很多需要注意的潜在问题。那么在这个文本框背后是什么呢？你可以在里面输入任何内容然后按回车键，但是……
(00:32) what should we be putting there and what are these words generated back how does this work and what what are you talking to exactly so I'm hoping to get at all those topics in this video we're going to go through the entire pipeline of how this stuff is built but I'm going to keep everything uh sort of accessible to a general audience so let's take a look at first how you build something like chpt and along the way I'm going to talk about um you know some of the sort of cognitive psychological implications of
我们应该在那里输入什么呢？生成回来的这些文字又是什么呢？它是如何工作的呢？以及你到底在和什么交流呢？所以我希望在这个视频中涵盖所有这些话题。我们将详细介绍这类东西是如何构建的整个流程，但我会让所有内容都能让普通大众理解。那么，让我们首先看看如何构建像 ChatGPT 这样的东西，并且在这个过程中，我会谈论一些关于这个工具在认知心理学方面的影响。
(01:00) the tools okay so let's build Chachi PT so there's going to be multiple stages arranged sequentially the first stage is called the pre-training stage and the first step of the pre-training stage is to download and process the internet now to get a sense of what this roughly looks like I recommend looking at this URL here so um this company called hugging face uh collected and created and curated this data set called Fine web and they go into a lot of detail on this block post on how how they constructed the fine web data set and
这个工具。好的，那么让我们来构建 ChatGPT。这将会有多个按顺序排列的阶段。第一个阶段叫做预训练阶段，预训练阶段的第一步是下载并处理互联网上的内容。现在，为了大致了解这是什么样子的，我建议看看这里的这个网址。所以，有一家叫做 Hugging Face 的公司收集、创建并整理了这个叫做 “Fine web” 的数据集，他们在这篇博客文章中详细介绍了他们是如何构建 “Fine web” 数据集的。
(01:32) all of the major llm providers like open AI anthropic and Google and so on will have some equivalent internally of something like the fine web data set so roughly what are we trying to achieve here we're trying to get ton of text from the internet from publicly available sources so we're trying to have a huge quantity of very high quality documents and we also want very large diversity of documents because we want to have a lot of knowledge inside these models so we want large diversity of high quality documents and we want
所有主要的大语言模型提供商，比如 OpenAI、Anthropic 和谷歌等等，在内部都会有类似于 “Fine web” 数据集的东西。那么，大致来说，我们在这里想要实现什么呢？我们试图从互联网上的公开可用资源中获取大量的文本。所以我们试图拥有大量非常高质量的文档，并且我们也希望文档具有很大的多样性，因为我们希望这些模型中包含很多知识。所以我们想要大量多样的高质量文档，并且我们想要……
(02:02) many many of them and achieving this is uh quite complicated and as you can see here takes multiple stages to do well so let's take a look at what some of these stages look like in a bit for now I'd like to just like to note that for example the fine web data set which is fairly representative what you would see in a production grade application actually ends up being only about 44 terabyt of dis space um you can get a USB stick for like a terabyte very easily or I think this could fit on a single hard drive almost today so this
很多很多这样的文档。而实现这一点是相当复杂的，并且正如你在这里看到的，要做好这件事需要多个阶段。所以让我们稍后看看这些阶段中的一些是什么样子的。现在，我只是想指出，例如，“Fine web” 数据集在生产级应用中是相当有代表性的，实际上它最终只占用大约 44 太字节的磁盘空间。你可以很容易地买到一个 1 太字节的 U 盘，或者我认为如今这个数据量几乎可以装在一个单个的硬盘上。所以这个……
(02:29) is not a huge amount of data at the end of the day even though the internet is very very large we're working with text and we're also filtering it aggressively so we end

up with about 44 terabytes in this example so let's take a look at uh kind of what this data looks like and what some of these stages uh also are so the starting point for a lot of these efforts and something that contributes most of the data by the end of it is Data from common crawl so common craw is an organization that has been basically
归根结底并不是大量的数据。尽管互联网非常非常庞大，但我们处理的是文本，并且我们也在大力筛选这些文本。所以在这个例子中，我们最终得到了大约 44 太字节的数据。那么，让我们看看这种数据是什么样子的，以及这些阶段中的一些都是什么。所以，很多这些工作的起点，并且到最后贡献了大部分数据的是来自 Common Crawl 的数据。Common Crawl 基本上是一个组织……
(02:57) scouring the internet since 2007 so as of 2024 for example common CW has indexed 2.7 billion web pages uh and uh they have all these crawlers going around the internet and what you end up doing basically is you start with a few seed web pages and then you follow all the links and you just keep following links and you keep indexing all the information and you end up with a ton of data of the internet over time so this is usually the starting point for a lot of the uh for a lot of these efforts now this common C
自 2007 年以来一直在搜索互联网。例如，截至 2024 年，Common Crawl 已经索引了 27 亿个网页。并且他们有所有这些网络爬虫在互联网上运行。基本上，你最终做的事情是，从一些种子网页开始，然后跟踪所有的链接，你不断地跟踪链接，并不断地索引所有的信息，随着时间的推移，你最终会得到大量的互联网数据。所以这通常是很多这些工作的起点。现在，这个 Common Crawl……
(03:26) data is quite raw and is filtered in many many different ways so here they Pro they document this is the same diagram they document a little bit the kind of processing that happens in these stages so the first thing here is something called URL filtering so what that is referring to is that there's these block lists of uh basically URLs that are or domains that uh you don't want to be getting data from so usually this includes things like U malware websites spam websites marketing websites uh racist websites adult sites and things
数据相当原始，并且要通过许多不同的方式进行筛选。所以在这里，他们展示了（文档中）这是同样的图表，他们稍微记录了一下在这些阶段发生的那种处理过程。所以这里的第一件事叫做 URL 筛选。这指的是有这些黑名单，基本上是你不想从中获取数据的 URL 或域名。所以通常这包括像恶意软件网站、垃圾邮件网站、营销网站、种族主义网站、成人网站之类的东西。
(04:01) like that so there's a ton of different types of websites that are just eliminated at this stage because we don't want them in our data set um the second part is text extraction you have to remember that all these web pages this is the raw HTML of these web pages that are being saved by these crawlers so when I go to inspect here this is what the raw HTML actually looks like you'll notice that it's got all this markup uh like lists and stuff like that and there's CSS and all this kind of stuff so this is um computer
诸如此类。所以有大量不同类型的网站在这个阶段就被剔除了，因为我们不希望它们在我们的数据集中。第二部分是文本提取。你必须记住，所有这些网页，这是这些爬虫保存的这些网页的原始 HTML 代码。所以当我在这里检查时，这就是原始 HTML 实际上的样子。你会注意到它有所有这些标记，比如列表之类的东西，还有 CSS 以及所有这类东西。所以这是…… 计算机……
(04:31) code almost for these web pages but what we really want is we just want this text right we just want the text of this web page and we don't want the navigation and things like that so there's a lot of filtering and processing uh and heris that go into uh adequately filtering for just their uh good content of these web pages the next stage here is language filtering so for example fine web filters uh using a language classifier they try to guess what language every single web page is in and then they only keep web pages that have more than 65%
几乎是这些网页的代码，但我们真正想要的是，我们只想要这些文本，对吧？我们只想要这个网页的文本内容，而不想要导航栏之类的东西。所以有很多筛选和处理工作，以及一些启发式方法，用于充分筛选出这些网页的优质内容。这里的下一个阶段是语言筛选。例如，“Fine web”使用语言分类器进行筛选，他们试图猜测每个网页的语言是什么，然后他们只保留英语含量超过 65% 的网页（举个例子）。
(05:03) of English as an example and so you can get a sense that this is like a design decision that different companies can uh can uh take for themselves what fraction of all different types of languages are we going to include in our data set because for example if we filter out all of the Spanish as an example then you might imagine that our model later will not be very good at Spanish because it's just never seen that much data of that language and so different companies can focus on multilingual performance to uh
作为一个例子。所以你可以感觉到，这是一个不同公司可以自行做出的设计决策，即我们要在数据集中包含各种不同语言的多大比例。因为例如，如果我们把所有西班牙语的内容都筛选掉，那么你可以想象，我们的模型后来在处理西班牙语时可能就不会很好，因为它从来没有见过那么多那种语言的数据。所以不同的公司可以在不同程度上关注多语言性能。
(05:29) to a different degree as an example so fine web is quite focused on English and so

their language model if they end up training one later will be very good at English but not may be very good at other languages after language filtering there's a few other filtering steps and D duplication and things like that um finishing with for example the pii removal this is personally identifiable information so as an example addresses Social Security numbers and things like that you would try to detect them and you would try to filter out those kinds

举个例子。所以 “Fine web” 相当关注英语，因此他们的语言模型，如果他们后来最终训练出一个的话，在英语方面会非常好，但在其他语言方面可能就不会那么好。在语言筛选之后，还有一些其他的筛选步骤，比如去重等等。最后例如会进行个人可识别信息（PII）的去除。这就是像地址、社保号码之类的个人可识别信息。你会试图检测到这些信息，并试图从数据集中筛选出包含这类信息的网页。

(05:59) of web pages from the the data set as well so there's a lot of stages here and I won't go into full detail but it is a fairly extensive part of the pre-processing and you end up with for example the fine web data set so when you click in on it uh you can see some examples here of what this actually ends up looking like and anyone can download this on the huging phase web page and so here are some examples of the final text that ends up in the training set so this is some article about tornadoes in 2012 um so there's some t tadoes in 2020

也从数据集中筛选出去。所以这里有很多阶段，我不会详细介绍所有细节，但这是预处理中相当广泛的一部分。最终你会得到例如 “Fine web” 数据集。所以当你点击进去时，你可以在这里看到一些它最终实际样子的例子，并且任何人都可以在 Hugging Face 网站上下载这个数据集。所以这里有一些最终进入训练集的文本示例。这是一篇关于 2012 年龙卷风的文章。所以在 2020 年有一些…… 龙卷风……

(06:31) in 2012 and what happened uh this next one is something about did you know you have two little yellow 9vt battery sized adrenal glands in your body okay so this is some kind of a odd medical article so just think of these as basically uh web pages on the internet filtered just for the text in various ways and now we have a ton of text 40 terabytes off it and that now is the starting point for the next step of this stage now I wanted to give you an intuitive sense of where we are right now so I took the first 200 web pages

在 2012 年以及发生了什么。下一个是关于 “你知道你身体里有两个像 9 伏电池大小的黄色肾上腺吗” 之类的内容。好的，所以这是某种奇怪的医学文章。所以就把这些基本上看作是互联网上的网页，经过了各种方式的文本筛选。现在我们有了大量的文本，40 太字节的文本，而这现在是这个阶段下一步的起点。现在我想让你直观地了解我们目前处于什么阶段。所以我拿了前 200 个网页。

(07:07) here and remember we have tons of them and I just take all that text and I just put it all together concatenate it and so this is what we end up with we just get this just just raw text raw internet text and there's a ton of it even in these 200 web pages so I can continue zooming out here and we just have this like massive tapestry of Text data and this text data has all these p patterns and what we want to do now is we want to start training neural networks on this data so the neural networks can internalize and model how this text

这里，记住我们有大量的网页，我把所有那些文本都拿出来，然后把它们全部拼接在一起。所以这就是我们最终得到的，我们得到的就是这些原始文本，原始的互联网文本。即使在这 200 个网页中也有大量的文本。所以我可以继续在这里缩小视图，我们就有了这样一个巨大的文本数据织锦。这些文本数据有所有这些模式，而我们现在想做的是，我们想开始在这些数据上训练神经网络，这样神经网络就可以内化并模拟这些文本是如何流动的。

(07:39) flows right so we just have this giant texture of text and now we want to get neural Nets that mimic it okay now before we plug text into neural networks we have to decide how we're going to represent this text uh and how we're going to feed it in now the way our technology works for these neuron Lots is that they expect a one-dimensional sequence of symbols and they want a finite set of symbols that are possible and so we have to decide what are the symbols and then we have to represent our data as one-dimensional sequence of those

对吧？所以我们有了这个巨大的文本结构，现在我们想要得到能够模仿它的神经网络。好的，现在在我们把文本输入到神经网络之前，我们必须决定如何表示这些文本，以及如何将其输入。现在，我们这些神经网络的技术原理是，它们期望的是一个一维的符号序列，并且它们需要一组有限的可能符号。所以我们必须决定这些符号是什么，然后我们必须把我们的数据表示为这些符号的一维序列。

(08:14) symbols so right now what we have is a onedimensional sequence of text it starts here and it goes here and then it comes here Etc so this is a onedimensional sequence even though on my monitor of course it's laid out in a two-dimensional way but it goes from left to right and top to bottom right so it's a one-dimensional sequence of text now this being computers of course there's an underlying representation here so if I do what's called utf8 uh encode this text then I can get the raw bits that correspond to this text in the

符号。所以现在，我们拥有的是一个文本的一维序列，它从这里开始，然后到这里，然后到这里等等。所以这是一个一维序列，尽管在我的显示器上当然是以二维方式呈现的，但它是从左到右、从上到下的。所以这是一个文本的一维序列。现在，因为这是在计算机上，当然这里有一个底层表示。所以如果我进行所谓的 UTF-8 编码这个文本，那么我可以得到在计算机中与这个文本相对应的原始位。

(08:44) computer and that's what uh that looks like this so it turns out that for example this very first bar here is the first uh eight bits as an example so what is this thing right this is um representation that we are looking for uh in in a certain sense we have exactly two possible symbols zero and one and we have a very long sequence of it right now as it turns out um this sequence length is actually going to be very finite and precious resource uh in our neural network and we actually don't want extremely long sequences of just

计算机中的样子就是这样。所以结果是，例如，这里的第一个条形图就是第一个 8 位作为一个例子。那么这是什么东西呢？这就是…… 我们正在寻找的一种表示。在某种意义上，我们恰好有两个可能的符号，0 和 1，并且我们现在有一个非常长的这样的序列。结果是，这个序列长度实际上在我们的神经网络中是非常有限且宝贵的资源，并且我们实际上不想要仅仅由两个符号组成的极其长的序列。

(09:23) two symbols instead what we want is we want to trade off uh this um symbol size uh of this vocabulary as we call it and the resulting sequence length so we don't want just two symbols and extremely long sequences we're going to want more symbols and shorter sequences okay so one naive way of compressing or decreasing the length of our sequence here is to basically uh consider some group of consecutive bits for example eight bits and group them into a single what's called bite so because uh these bits are either on or off if we take a

而不是两个符号。我们想要的是在我们所说的词汇表的符号大小和由此产生的序列长度之间进行权衡。所以我们不想要只有两个符号和极长的序列，我们想要更多的符号和更短的序列。好的，那么一种简单的压缩或缩短我们这里序列长度的方法是，基本上考虑一些连续的位，例如 8 位，并将它们组合成一个所谓的字节。因为这些位要么是开要么是关，如果我们取一组 8 位的……

(10:00) group of eight of them there turns out to be only 256 possible combinations of how these bits could be on or off and so therefore we can re - represent this sequence into a sequence of bytes instead so this sequence of bytes will be eight times shorter but now we have 256 possible symbols so every number here goes from 0 to 255 now I really encourage you to think of these not as numbers but as unique IDs or like unique symbols so maybe it's a bit more maybe it's better to actually think of these to replace every one of

它们，结果发现这些位的开关组合只有 256 种可能。因此，我们可以将这个序列重新表示为一个字节序列。所以这个字节序列将缩短 8 倍，但现在我们有 256 种可能的符号。所以这里的每个数字都从 0 到 255。现在我真的建议你不要把这些看作数字，而是看作唯一的 ID 或独特的符号。也许更恰当的是，最好把这些想象成用独特的表情符号来替换每一个……

(10:32) these with a unique Emoji you'd get something like this so um we basically have a sequence of emojis and there's 256 possible emojis you can think of it that way now it turns out that in production for state - of - the - art language models uh you actually want to go even Beyond this you want to continue to shrink the length of the sequence uh because again it is a precious resource in return for more symbols in your vocabulary and the way this is done is done by running what's called The Bite pair encoding algorithm and the way this

符号，你会得到类似这样的东西。所以，我们基本上有一个表情符号序列，并且有 256 种可能的表情符号，你可以这样去理解。现在事实证明，在生产最先进的语言模型时，实际上你想要更进一步。你想要继续缩短序列的长度，因为它仍然是一种宝贵的资源，作为在词汇表中拥有更多符号的交换。实现这一点的方法是运行所谓的字节对编码算法，其工作方式是……

(11:04) works is we're basically looking for consecutive bytes or symbols that are very common so for example turns out that the sequence 116 followed by 32 is quite common and occurs very frequently so what we're going to do is we're going to group uh this um pair into a new symbol so we're going to Mint a symbol with an ID 256 and we're going to rewrite every single uh pair 11632 with this new symbol and then can we can iterate this algorithm as many times as we wish and each time when we mint a new symbol we're decreasing the length and

我们基本上在寻找非常常见的连续字节或符号。例如，结果发现 116 后面跟着 32 的序列相当常见，并且出现得非常频繁。所以我们要做的是，将这个对组合成一个新符号。我们将创建一个 ID 为 256 的符号，并且我们将用这个新符号重写每一个 11632 对。然后我们可以根据需要多次迭代这个算法，每次我们创建一个新符号时，我们都在缩短序列长度，并且……

(11:41) we're increasing the symbol size and in practice it turns out that a pretty good setting of um the basically the vocabulary size turns out to be about 100,000 possible symbols so in particular GPT 4 uses 100, 277 symbols um and this process of converting from raw text into these symbols or as we call them tokens is the process called tokenization so let's now take a look at how gp4 performs tokenization conting from text to tokens and from tokens back to text and what this actually looks like so one website I like to use to

增加符号数量。在实践中，结果发现一个相当不错的设置是，基本上词汇表大小约为 100,000 个可能的符号。特别是 GPT 4 使用 100,277 个符号。从原始文本转换为这些符号（我们称之为标记）的过程称为标记化。那么现在让我们看看 GPT 4 是如何进行标记化的，从文本到标记，再从标记回到文本，实际情况是怎样的。我喜欢使用一个网站来……

(12:22) explore these token representations is called tick tokenizer and so come here to the drop down and select CL 100 a base which is the gp4 base model tokenizer and here on the left you can put in text and it shows you the tokenization of that text so for example heo space world so hello world turns out to be exactly two Tokens The Token hello which is the token with ID 15339 and the token space world that is the token 1 1917 so um hello space world now if I was to join these two for example I'm going to get again two tokens but it's

探索这些标记表示，这个网站叫做 tiktokenizer。来到这里的下拉菜单，选择 CL 100 a base，这是 GPT 4 基础模型的标记器。在左边你可以输入文本，它会显示该文本的标记化结果。例如，“heo space world”，结果“hello world” 恰好是两个标记，“hello” 标记的 ID 是 15339，“space world” 标记的 ID 是 11917。所以，“hello space world”。现在，如果我把这两个连接起来，例如，我又会得到两个标记，但它是……

(13:06) the token H followed by the token L world without the H um if I put in two Spa two spaces here between hello and world it's again a different uh tokenization there's a new token 220 here okay so you can play with this and see what happens here also keep in mind this is not uh this is case sensitive so if this is a capital H it is something else or if it's uh hello world then actually this ends up being three tokens instead of just two tokens yeah so you can play with this and get an sort of like an intuitive sense of uh what these tokens work like

“h” 标记后面跟着 “l world” 标记（没有 “h”）。如果我在 “hello” 和 “world” 之间输入两个空格，这又是一种不同的标记化方式，这里会出现一个新的标记 220。好的，所以你可以试试这个，看看会发生什么。也要记住，这是区分大小写的。所以如果是大写的 “H”，那就是另外一回事了，或者如果是 “hello world”（中间没有空格），实际上这最终会是三个标记，而不是两个标记。是的，所以你可以试试这个，对这些标记的工作方式有一个直观的感受。

(13:47) we're actually going to loop around to tokenization a bit later in the video for now I just wanted to show you the website and I wanted to uh show you that this text basically at the end of the day so for example if I take one line here this is what GT4 will see it as so this text will be a sequence of length 62 this is the sequence here and this is how the chunks of text correspond to these symbols and again there's 100, 27777 possible symbols and we now have one - dimensional sequences of those symbols so um yeah we're going to come

在视频后面我们还会再回到标记化这个话题。现在我只是想给你展示这个网站，并且想给你展示，归根结底，例如，如果我取这里的一行文本，这就是 GPT 4 会看到的内容。这段文本将是一个长度为 62 的序列，这是这里的序列，这是文本块与这些符号的对应方式。同样，有 100,277（此处原内容疑似多写了一个 7）个可能的符号，我们现在有这些符号的一维序列。所以，是的，我们会再……

(14:24) back to tokenization but that's uh for now where we are okay so what I've done now is I've taken this uh sequence of text that we have here in the data set and I have re - represented it using our tokenizer into a sequence of tokens and this is what that looks like now so for example when we go back to the Fine web data set they mentioned that not only is this 44 terab of dis space but this is about a 15 trillion token sequence of um in this data set and so here these are just some of the first uh one or two or

回到标记化这个话题，但目前我们先讲到这里。好的，所以我现在所做的是，我拿了我们数据集中的这个文本序列，并用我们的标记器将其重新表示为一个标记序列，现在这就是它的样子。例如，当我们回到 “Fine web” 数据集时，他们提到，这个数据集不仅占用 44 太字节的磁盘空间，而且其中大约有一个 15 万亿标记的序列。所以这里只是其中的一些开头的…… 一两个或……

(14:56) three or a few thousand here I think uh tokens of this data set but there's 15 trillion here uh to keep in mind and again keep in mind one more time that all of these represent little text chunks they're all just like atoms of these sequences and the numbers here don't make any sense they're just uh they're just unique IDs okay so now we get to the fun part which is the uh neural network training and this is where a lot of the heavy lifting happens computationally when you're training these neural networks so what we do here

三或几千个标记，我想，这是这个数据集的标记。但要记住这里有 15 万亿个标记。再提醒一次，要记住所有这些都代表小的文本块，它们就像是这些序列的原子。这里的数字本身没有意义，它们只是…… 只是唯一的 ID。好的，

所以现在我们来到了有趣的部分，也就是神经网络训练。当你训练这些神经网络时，这里在计算方面有很多繁重的工作。所以我们在这里做的是……

(15:28) in this this step is we want to model the statistical relationships of how these tokens follow each other in the sequence so what we do is we come into the data and we take Windows of tokens so we take a window of tokens uh from this data fairly randomly and um the windows length can range anywhere anywhere between uh zero tokens actually all the way up to some maximum size that we decide on uh so for example in practice you could see a token with Windows of say 8,000 tokens now in principle we can use arbitrary

在这一步中，我们想要对这些标记在序列中如何相互跟随的统计关系进行建模。所以我们所做的是，进入数据并选取标记窗口。我们从这个数据中相当随机地选取一个标记窗口，窗口长度实际上可以在 0 个标记到我们决定的某个最大尺寸之间的任何地方。例如，在实践中，你可能会看到一个窗口大小为 8000 个标记的情况。原则上，我们可以使用任意……

(16:04) window lengths of tokens uh but uh processing very long uh basically U window sequences would just be very computationally expensive so we just kind of decide that say 8,000 is a good number or 4,000 or 16,000 and we crop it there now in this example I'm going to be uh taking the first four tokens just so everything fits nicely so these tokens we're going to take a window of four tokens this bar view in and space single which are these token IDs and now what we're trying to do here is we're trying to basically predict the

长度的标记窗口，但处理非常长的窗口序列在计算上会非常昂贵。所以我们只是决定，比如说 8000 是个不错的数字，或者 4000、16000，然后我们在那里截断。在这个例子中，我将选取前四个标记，这样一切都能很好地展示。所以我们选取这四个标记作为一个窗口，“bar view in” 和 “space single”，这些是它们的标记 ID。现在我们在这里试图做的是，基本上预测……

(16:43) token that comes next in the sequence so 3962 comes next right so what we do now here is that we call this the context these four tokens are context and they feed into a neural network and this is the input to the neural network now I'm going to go into the detail of what's inside this neural network in a little bit for now it's important to understand is the input and the output of the neural net so the input are sequences of tokens of variable length anywhere between zero and some maximum size like 8,000 the output now is a

序列中的下一个标记。所以下一个是 3962，对吧？所以我们现在在这里所做的是，把这称为上下文，这四个标记就是上下文，它们被输入到神经网络中，这就是神经网络的输入。现在我稍后会详细介绍这个神经网络的内部结构。目前重要的是要理解神经网络的输入和输出。输入是长度可变的标记序列，长度在 0 到某个最大尺寸（比如 8000）之间。现在的输出是……

(17:18) prediction for what comes next so because our vocabulary has 100277 possible tokens the neural network is going to Output exactly that many numbers and all of those numbers correspond to the probability of that token as coming next in the sequence so it's making guesses about what comes next um in the beginning this neural network is randomly initialized so um and we're going to see in a little bit what that means but it's a it's a it's a random transformation so these probabilities in the very beginning of the training are also going to be kind

对下一个出现的标记的预测。因为我们的词汇表有 100277 个可能的标记，神经网络将输出恰好这么多个数字，所有这些数字对应于该标记作为序列中下一个出现的概率。所以它在猜测下一个会出现什么。一开始，这个神经网络是随机初始化的。所以…… 我们稍后会看到这意味着什么，但这是一个随机变换。所以在训练刚开始时，这些概率也会有点……

(17:51) of random uh so here I have three examples but keep in mind that there's 100,000 numbers here um so the probability of this token space Direction neural network is saying that this is 4% likely right now 11799 is 2% and then here the probility of 3962 which is post is 3% now of course we've sampled this window from our data set so we know what comes next we know and that's the label we know that the correct answer is that 3962 actually comes next in the sequence so now what we have is this mathematical process for

随机。所以这里我有三个例子，但要记住这里有 100,000 个数字。所以这个 “space Direction” 标记，神经网络说它现在有 4% 的可能性出现，11799 这个标记有 2% 的可能性，然后这里 3962（“post”）这个标记有 3% 的可能性。当然，我们从数据集中采样了这个窗口，所以我们知道接下来是什么，我们知道，那就是标签。我们知道正确答案是 3962 实际上是序列中的下一个标记。所以现在我们有这个数学过程来……

(18:26) doing an update to the neural network we have the way of tuning it and uh we're going to go into a little bit of of detail in a bit but basically we know that this probability here of 3% we want this probability to be higher and we want the probabilities of all the other tokens to be lower and so we have a way of mathematically calculating how to adjust and update the neural network so that the correct answer has a slightly higher probability so if I do an update to the neural network now the next time I Fe this

particular sequence of four tokens
对神经网络进行更新，我们有调整它的方法，一会儿我们会深入探讨一些细节。但基本上，我们知道这里 3% 的概率，我们希望这个概率更高，同时希望所有其他标记的概率更低。所以我们有数学计算方法来调整和更新神经网络，让正确答案的概率稍微提高。如果我现在对神经网络进行更新，下次我输入这四个标记的特定序列时，
(19:01) into neural network the neural network will be slightly adjusted now and it will say Okay post is maybe 4% and case now maybe is 1% and uh Direction could become 2% or something like that and so we have a way of nudging of slightly updating the neuronet to um basically give a higher probability to the correct token that comes next in the sequence and now you just have to remember that this process happens not just for uh this um token here where these four fed in and predicted this one this process happens at the same time for all of these tokens
神经网络就会稍微调整，它可能会说 “post” 的概率现在是 4%，“case” 的概率现在可能是 1%，“Direction” 的概率可能变成 2% 之类的。所以我们有一种微调的方法，稍微更新神经网络，基本上是给序列中下一个正确的标记更高的概率。现在你要记住，这个过程不仅仅发生在这四个标记输入并预测这一个标记的情况，而是对数据集中的所有标记同时进行。
(19:36) in the entire data set and so in practice we sample little windows little batches of Windows and then at every single one of these tokens we want to adjust our neural network so that the probability of that token becomes slightly higher and this all happens in parallel in large batches of these tokens and this is the process of training the neural network it's a sequence of updating it so that it's predictions match up the statistics of what actually happens in your training set and its probabilities become consistent with the uh statistical
在整个数据集中。所以在实践中，我们采样小窗口，一小批一小批的窗口，然后对每个标记，我们都要调整神经网络，使该标记的概率稍微提高。这一切都是在大量标记的批次中并行进行的，这就是训练神经网络的过程。这是一个更新的过程，让它的预测与训练集中实际发生的统计数据相匹配，并且它的概率与数据集中标记之间相互跟随的统计模式一致。
(20:08) patterns of how these tokens follow each other in the data so let's now briefly get into the internals of these neural networks just to give you a sense of what's inside so neural network internals so as I mentioned we have these inputs uh that are sequences of tokens in this case this is four input tokens but this can be anywhere between zero up to let's say 8,000 tokens in principle this can be an infinite number of tokens we just uh it would just be too computationally expensive to process an infinite number of tokens so we just
模式一致。那么现在让我们简要了解一下这些神经网络的内部结构，让你对其内部有个概念。神经网络内部结构：正如我提到的，我们有这些输入，它们是标记序列。在这种情况下，这是四个输入标记，但实际上它可以在 0 到比如说 8000 个标记之间。原则上，它可以是无限数量的标记，但处理无限数量的标记在计算上成本太高，所以我们只是……
(20:38) crop it at a certain length and that becomes the maximum context length of that uh model now these inputs X are mixed up in a giant mathematical expression together with the parameters or the weights of these neural networks so here I'm showing six example parameters and their setting but in practice these uh um modern neural networks will have billions of these uh parameters and in the beginning these parameters are completely randomly set now with a random setting of parameters you might expect that this uh this neural network
在某个长度截断，这就成为了那个模型的最大上下文长度。现在这些输入 X 与这些神经网络的参数或权重在一个巨大的数学表达式中混合在一起。这里我展示了六个示例参数及其设置，但实际上，这些现代神经网络会有数十亿个这样的参数。一开始，这些参数是完全随机设置的。现在，由于参数是随机设置的，你可能会认为这个神经网络……
(21:14) would make random predictions and it does in the beginning it's totally random predictions but it's through this process of iteratively updating the network uh as and we call that process training a neural network so uh that the setting of these parameters gets adjusted such that the outputs of our neural network becomes consistent with the patterns seen in our training set so think of these parameters as kind of like knobs on a DJ set and as you're twiddling these knobs you're getting different uh predictions for every
会做出随机预测，一开始确实是这样，它的预测完全是随机的。但通过这个迭代更新网络的过程，我们称之为训练神经网络，这些参数的设置会被调整，使得我们神经网络的输出与训练集中看到的模式一致。把这些参数想象成 DJ 设备上的旋钮，当你转动这些旋钮时，对于每一个可能的标记序列输入，你都会得到不同的预测。
(21:45) possible uh token sequence input and training in neural network just means discovering a setting of parameters that seems to be consistent with the statistics of the training set now let me just give you an example what this giant mathematical expression

looks like just to give you a sense and modern networks are massive expressions with trillions of terms probably but let me just show you a simple example here it would look something like this I mean these are the kinds of Expressions just to show you that it's not very scary we

可能的标记序列输入。训练神经网络就意味着找到一组与训练集统计数据相符的参数设置。现在让我给你举个例子，展示一下这个巨大的数学表达式是什么样子，让你有个概念。现代网络的表达式可能包含数万亿项，但我在这里给你展示一个简单的例子。它看起来大概是这样，我的意思是这些表达式是为了让你知道它并没有那么可怕。我们……

(22:14) have inputs x uh like X1 x2 in this case two example inputs and they get mixed up with the weights of the network w0 W1 2 3 Etc and this mixing is simple things like multiplication addition addition exponentiation division Etc and it is the subject of neural network architecture research to design effective mathematical Expressions uh that have a lot of uh kind of convenient characteristics they are expressive they're optimizable they're paralyzable Etc and so but uh at the end of the day these are these are not complex

有输入 x，比如这里的 X1、X2，这是两个示例输入，它们与网络的权重 w0、W1、2、3 等等混合在一起。这种混合是像乘法、加法、指数运算、除法等简单的运算。设计有效的数学表达式是神经网络架构研究的课题，这些表达式具有很多方便的特性，比如有表现力、可优化、可并行化等等。但归根结底，这些……

(22:50) expressions and basically they mix up the inputs with the parameters to make predictions and we're optimizing uh the parameters of this neural network so that the predictions come out consistent with the training set now I would like to show you an actual production grade example of what these neural networks look like so for that I encourage you to go to this website that has a very nice visualization of one of these networks so this is what you will find on this website and this neural network here that is used in production settings

表达式并不复杂，基本上就是将输入和参数混合起来进行预测。我们正在优化这个神经网络的参数，使预测结果与训练集一致。现在我想给你展示一个实际生产级别的神经网络示例，看看它们是什么样子。为此，我建议你访问这个网站，它对其中一个网络有非常好的可视化展示。这是你在这个网站上会看到的内容，这个在生产环境中使用的神经网络……

(23:22) has this special kind of structure this network is called the Transformer and this particular one as an example has 8 5,000 roughly parameters now here on the top we take the inputs which are the token sequences and then information flows through the neural network until the output which here are the logit softmax but these are the predictions for what comes next what token comes next and then here there's a sequence of Transformations and all these intermediate values that get produced inside this mathematical expression s it

有一种特殊的结构，这个网络叫做 Transformer，这个特定的网络大约有 85000 个参数。现在在顶部，我们输入标记序列，然后信息在神经网络中流动，直到输出，这里的输出是 logit softmax，这些是对下一个出现的标记的预测。然后这里有一系列的变换，以及在这个数学表达式中产生的所有中间值，它……

(23:59) is sort of predicting what comes next so as an example these tokens are embedded into kind of like this distributed representation as it's called so every possible token has kind of like a vector that represents it inside the neural network so first we embed the tokens and then those values uh kind of like flow through this diagram and these are all very simple mathematical Expressions individually so we have layer norms and Matrix multiplications and uh soft Maxes and so on so here kind of like the attention block of this Transformer and

有点像是在预测下一个出现的内容。例如，这些标记被嵌入到一种所谓的分布式表示中，每个可能的标记在神经网络内部都有一个向量来表示它。所以首先我们嵌入标记，然后这些值有点像在这个图表中流动，这些单独来看都是非常简单的数学表达式。我们有层归一化、矩阵乘法、softmax 等等。这里有点像这个 Transformer 的注意力模块，然后……

(24:32) then information kind of flows through into the multi - layer perceptron block and so on and all these numbers here these are the intermediate values of the expression and uh you can almost think of these as kind of like the firing rates of these synthetic neurons but I would caution you to uh not um kind of think of it too much like neurons because these are extremely simple neurons compared to the neurons you would find in your brain your biological neurons are very complex dynamical processes that have memory and so on

信息有点像流入多层感知机模块等等。这里所有这些数字是表达式的中间值，你几乎可以把这些看作是这些合成神经元的 firing rates（激发率），但我要提醒你，不要把它想得太像神经元，因为与你大脑中的神经元相比，这些是极其简单的神经元。你的生物神经元是非常复杂的动态过程，具有记忆等等。

(25:01) there's no memory in this expression it's a fixed mathematical expression from input to Output with no memory it's just a stateless so these are very simple neurons in comparison to biological neurons but you can still kind of loosely think of this as like a synthetic piece of uh brain tissue if you if you like uh to think about it that way so information flows through all these neurons fire until we get to the predictions now I'm not actually going to dwell too much on the precise kind of like mathematical details of all

这个表达式中没有记忆，它是一个从输入到输出的固定数学表达式，没有记忆，只是无状态的。所以与生物神经元相比，这些是非常简单的神经元。但如果你愿意，你仍然可以大致把它想象成一块合成的脑组织。所以信息在这些神经元中流动、激发，直到我们得到预测结果。现在我实际上不会过多地纠缠于所有这些变换的精确数学细节，老实说，我认为深入了解这些并不是那么重要。

(25:30) these Transformations honestly I don't think it's that important to get into what's really important to understand is that this is a mathematical function it is uh parameterized by some fixed set of parameters like say 85,000 of them and it is a way of transforming inputs into outputs and as we twiddle the parameters we are getting uh different kinds of predictions and then we need to find a good setting of these parameters so that the predictions uh sort of match up with the patterns seen in training set so that's the Transformer okay so I've

真正重要的是要理解这是一个数学函数，它由一组固定的参数参数化，比如说 85000 个参数，它是一种将输入转换为输出的方式。当我们调整参数时，我们会得到不同类型的预测。然后我们需要找到这些参数的一个好的设置，使预测结果与训练集中看到的模式相匹配。这就是 Transformer。好的，所以我已经……

(26:02) shown you the internals of the neural network and we talked a bit about the process of training it I want to cover one more major stage of working with these networks and that is the stage called inference so in inference what we're doing is we're generating new data from the model and so uh we want to basically see what kind of patterns it has internalized in the parameters of its Network so to generate from the model is relatively straightforward we start with some tokens that are basically your prefix like what you want

向你展示了神经网络的内部结构，并且我们也稍微讨论了训练它的过程。我想介绍使用这些网络的另一个主要阶段，那就是推理阶段。在推理阶段，我们要从模型中生成新的数据，所以我们基本上想看看它在网络参数中内化了什么样的模式。从模型生成数据相对来说比较直接，我们从一些标记开始，这些标记基本上就是你的前缀，比如你想要……

(26:33) to start with so say we want to start with the token 91 well we feed it into the network and remember that the network gives us probabilities right it gives us this probability Vector here so what we can do now is we can basically flip a biased coin so um we can sample uh basically a token based on this probability distribution so the tokens that are given High probability by the model are more likely to be sampled when you flip this biased coin you can think of it that way so we sample from the distribution to get a single unique

开始的内容。比如说我们想从标记 91 开始，我们把它输入到网络中，记住网络会给我们概率，对吧？它给我们这里的这个概率向量。所以我们现在可以做的是，基本上就像抛一枚有偏向的硬币。所以我们可以根据这个概率分布采样一个标记。模型赋予高概率的标记在抛这枚有偏向的硬币时更有可能被采样，你可以这样理解。所以我们从分布中采样得到一个唯一的标记。

(27:08) token so for example token 860 comes next uh so 860 in this case when we're generating from model could come next now 860 is a relatively likely token it might not be the only possible token in this case there could be many other tokens that could have been sampled but we could see that 86c is a relatively likely token as an example and indeed in our training examp example here 860 does follow 91 so let's now say that we um continue the process so after 91 there's a60 we append it and we again ask what is the third token let's sample and

例如，接下来是标记 860。所以在这种情况下，当我们从模型生成数据时，860 可能会接下来出现。860 是一个相对有可能出现的标记，在这种情况下它可能不是唯一可能被采样的标记，可能还有很多其他标记，但我们可以看到 860 是一个相对有可能的标记。实际上，在我们这里的训练示例中，860 确实跟随在 91 后面。那么现在假设我们继续这个过程，91 后面是 860，我们把它添加进去，然后我们再问第三个标记是什么，让我们采样，假设……

(27:42) let's just say that it's 287 exactly as here let's do that again we come back in now we have a sequence of three and we ask what is the likely fourth token and we sample from that and get this one and now let's say we do it one more time we take those four we sample and we get this one and this 13659 uh this is not actually uh 3962 as we had before so this token is the token article uh instead so viewing a single article and so in this case we didn't exactly reproduce the sequence that we saw here in the training data so keep in

它恰好是 287，就像这里一样。让我们再做一次，我们现在有一个三个标记的序列，我们问第四个可能的标记是什

么，然后采样得到这个。现在假设我们再做一次，我们取这四个标记再采样，得到这个 13659。实际上，它不是我们之前的 3962，这个标记是 “article” 标记。所以是 “viewing a single article”。在这种情况下，我们并没有完全重现我们在训练数据中看到的序列，所以要记住……
(28:20) mind that these systems are stochastic they have um we're sampling and we're flipping coins and sometimes we lock out and we reproduce some like small chunk of the text and training set but sometimes we're uh we're getting a token that was not verbatim part of any of the documents in the training data so we're going to get sort of like remixes of the data that we saw in the training because at every step of the way we can flip and get a slightly different token and then once that token makes it in if you sample the next one and so on you very
这些系统是随机的，我们在采样，就像抛硬币一样。有时我们会碰巧重现训练集中的一小段文本，但有时我们得到的标记并不是训练数据中任何文档的逐字内容。所以我们得到的是训练数据的一种 “混音版”，因为在每一步我们都可能得到一个略有不同的标记，一旦这个标记被选中，如果你再采样下一个，等等，你很快就会……
(28:53) quickly uh start to generate token streams that are very different from the token streams that UR in the training documents so statistically they will have similar properties but um they are not identical to your training data they're kind of like inspired by the training data and so in this case we got a slightly different sequence and why would we get "article" you might imagine that "article" is a relatively likely token in the context of "bar viewing single" Etc and you can imagine that the word "article" followed this context window somewhere
很快就会开始生成与训练文档中的标记流非常不同的标记流。从统计角度来看，它们会有相似的属性，但它们与训练数据并不完全相同，它们有点像是受到训练数据的启发。在这种情况下，我们得到了一个略有不同的序列。为什么会得到 “article” 呢？你可以想象，在 “bar viewing single” 等语境中，“article” 是一个出现概率相对较高的标记。你可以想象，在训练文档中的某个地方，“article” 这个词跟在这样的上下文窗口之后。
(29:25) in the training documents uh to some extent and we just happen to sample it here at that stage so basically inference is just uh predicting from these distributions one at a time we continue feeding back tokens and getting the next one and we uh we're always flipping these coins and depending on how lucky or unlucky we get um we might get very different kinds of patterns depending on how we sample from these probability distributions so that's inference so in most common scenarios uh basically downloading the internet and
在一定程度上，我们只是碰巧在那个阶段采样到了它。所以基本上，推理就是从这些分布中一次一个地进行预测。我们不断反馈标记并获取下一个标记，我们一直在 “抛硬币”，根据我们的运气好坏，根据我们从这些概率分布中的采样方式，我们可能会得到非常不同的模式。这就是推理。在大多数常见场景中，基本上，下载互联网数据并……
(29:57) tokenizing it is is a pre - processing step you do that a single time and then uh once you have your token sequence we can start training networks and in Practical cases you would try to train many different networks of different kinds of uh settings and different kinds of arrangements and different kinds of sizes and so you''ll be doing a lot of neural network training and um then once you have a neural network and you train it and you have some specific set of parameters that you're happy with um then you can take the model and you can
对其进行标记化是一个预处理步骤，你只需要做一次。然后，一旦你有了标记序列，我们就可以开始训练网络。在实际情况中，你会尝试训练许多不同的网络，它们具有不同的设置、不同的排列方式和不同的规模。所以你会进行大量的神经网络训练。然后，一旦你有了一个神经网络，并且训练好了，有了你满意的一组特定参数，那么你就可以使用这个模型，并且……
(30:26) do inference and you can actually uh generate data from the model and when you're on chat GPT and you're talking with a model uh that model is trained and has been trained by open aai many months ago probably and they have a specific set of Weights that work well and when you're talking to the model all of that is just inference there's no more training those parameters are held fixed and you're just talking to the model sort of uh you're giving it some of the tokens and it's kind of completing token sequences and that's
进行推理，实际上你可以从模型中生成数据。当你使用 ChatGPT 与模型交谈时，那个模型已经经过训练，可能是 OpenAI 在几个月前训练好的，他们有一组效果很好的特定权重。当你与模型交谈时，所有这些都只是推理，不再进行训练，那些参数是固定的。你只是在与模型交谈，给它一些标记，它在完成标记序列，这就是……
(30:55) what you're seeing uh generated when you actually use the model on CH GPT so that model then just does inference alone so let's now look at an example of training an inference that is kind of concrete and gives you a sense of what this actually looks like uh when these models are trained now the example that I would like to work with and that I'm particularly fond of is that of opening eyes gpt2 so GPT uh stands for generatively pre - trained Transformer and this is the second iteration of the GPT series by open AI when you are talking

你在使用 ChatGPT 模型时看到的生成内容。所以那个模型只是在进行推理。现在让我们看一个训练和推理的具体例子，让你了解这些模型在训练时实际的情况。我想用的例子，也是我特别喜欢的例子，是 OpenAI 的 GPT - 2。GPT 代表生成式预训练变换器（Generatively Pre - trained Transformer），这是 OpenAI 的 GPT 系列的第二代。当你现在与 ChatGPT 交谈时……

(31:24) to chat GPT today the model that is underlying all of the magic of that interaction is GPT 4 so the fourth iteration of that series now gpt2 was published in 2019 by openi in this paper that I have right here and the reason I like gpt2 is that it is the first time that a recognizably modern stack came together so um all of the pieces of gpd2 are recognizable today by modern standards it's just everything has gotten bigger now I'm not going to be able to go into the full details of this paper of course because it is a

背后的模型是 GPT - 4，这是该系列的第四代。GPT - 2 是 OpenAI 在 2019 年发表的，就在我手头的这篇论文中。我喜欢 GPT - 2 的原因是，它首次将一套可识别的现代技术组合在一起。所以按照现代标准，GPT - 2 的所有部分在今天都能被认出来，只是现在一切都变得更庞大了。当然，我不会深入探讨这篇论文的所有细节，因为这是一篇……

(31:57) technical publication but some of the details that I would like to highlight are as follows gpt2 was a Transformer neural network just like you were just like the neural networks you would work with today it was it had 1.6 billion parameters right so these are the parameters that we looked at here it would have 1.

技术出版物，但我想强调的一些细节如下：GPT - 2 是一个 Transformer 神经网络，就像你现在使用的神经网络一样。它有 16 亿个参数，对吧？这些就是我们之前提到的参数，它有……

(32:17) 6 billion of them today modern Transformers would have a lot closer to a trillion or several hundred billion probably the maximum context length here was 1,24 tokens so it is when we are sampling chunks of Windows of tokens from the data set we're never taking more than 1,24 tokens and so when you are trying to predict the next token in a sequence you will never have more than 1,24 tokens uh kind of in your context in order to make that prediction now this is also tiny by modern standards today the token uh the context lengths would be a lot closer to um couple

16 亿个。如今，现代的 Transformer 可能有接近一万亿或几千亿个参数。这里的最大上下文长度是 124 个标记。所以当我们从数据集中采样标记窗口时，我们从不取超过 124 个标记。所以当你试图预测序列中的下一个标记时，在你的上下文中，为了做出这个预测，你永远不会有超过 124 个标记。按照现代标准，这也很小。如今，标记的上下文长度更接近…… 几百

(32:53) hundred thousand or maybe even a million and so you have a lot more context a lot more tokens in history history and you can make a lot better prediction about the next token in the sequence in that way and finally gpt2 was trained on approximately 100 billion tokens and this is also fairly small by modern standards as I mentioned the fine web data set that we looked at here the fine web data set has 15 trillion tokens uh so 100 billion is is quite small now uh I actually tried to reproduce uh gpt2 for fun as part of this project

千，甚至可能是一百万。所以你有更多的上下文，更多的历史标记，这样你就可以更好地预测序列中的下一个标记。最后，GPT - 2 是在大约 1000 亿个标记上进行训练的，按照现代标准，这也相当小。正如我提到的，我们这里看的 Fine Web 数据集有 15 万亿个标记，所以 1000 亿是相当小的。实际上，我作为这个叫做 lm.c 项目的一部分，为了好玩尝试复现了 GPT - 2。

(33:24) called lm. C so you can see my rup of doing that in this post on GitHub under the lm. C repository so in particular the cost of training gpd2 in 2019 what was estimated to be approximately $40,000 but today you can do significantly better than that and in particular here it took about one day and about $600 uh but this wasn't even trying too hard I think you could really bring this down to about $100 today now why is it that the costs have come down so much well number one these data sets have gotten a lot better and the way we

你可以在 GitHub 上 lm.c 仓库下的这篇文章中看到我复现的过程。特别是在 2019 年，训练 GPT - 2 的成本估计约为 4 万美元，但如今你可以做得比那好得多。特别是在这里，这花了大约一天时间和 600 美元，但这还不是很努力的结果。我认为如今你真的可以把成本降到大约 100 美元。为什么成本下降了这么多呢？第一，这些数据集变得好多了，而且我们……

(34:01) filter them extract them and prepare them has gotten a lot more refined and so the data set is of just a lot higher quality so that's one thing but really the biggest difference is that our computers have gotten much faster in terms of the hardware and we're going to look at that in a second and also the software for uh running these models and really squeezing out all all the speed from the hardware as it is possible uh that software has also gotten much better as as everyone has focused on these models and try to run them very

筛选、提取和准备数据集的方式也更加精细，所以数据集的质量更高了。这是一方面，但真正最大的区别是，我们

计算机的硬件速度变得快多了，我们马上会讲到这一点。而且运行这些模型、充分发挥硬件速度的软件也变得更好了，因为每个人都专注于这些模型，并试图让它们运行得非常快。
(34:30) very quickly now I'm not going to be able to go into the full detail of this gpd2 reproduction and this is a long technical post but I would like to still give you an intuitive sense for what it looks like to actually train one of these models as a researcher like what are you looking at and what does it look like what does it feel like so let me give you a sense of that a little bit okay so this is what it looks like let me slide this over so what I'm doing here is I'm training a gpt2 model right now and um what's happening here is that
现在我无法深入介绍 GPT - 2 复现的所有细节，这是一篇很长的技术文章。但我还是想让你直观地感受一下作为一名研究人员实际训练这样一个模型是什么样的，你在看什么，它看起来怎么样，感觉如何。让我稍微给你讲讲。好的，这就是它的样子。我把这个移过来。我现在正在训练一个 GPT - 2 模型，这里发生的事情是……
(35:01) every single line here like this one is one update to the model so remember how here we are um basically making the prediction better for every one of these tokens and we are updating these weights or parameters of the neural net so here every single line is One update to the neural network where we change its parameters by a little bit so that it is better at predicting next token and sequence in particular every single line here is improving the prediction on 1 million tokens in the training set so we've basically taken 1 million tokens
这里的每一行，就像这一行，都是对模型的一次更新。还记得我们在这里基本上是为每个标记改进预测，并且更新神经网络的权重或参数吗？所以这里的每一行都是对神经网络的一次更新，我们对其参数进行一点调整，使其在预测序列中的下一个标记时表现得更好。特别是这里的每一行都在改进对训练集中 100 万个标记的预测。所以我们基本上从训练集中选取了 100 万个标记……
(35:39) out of this data set and we've tried to improve the prediction of that token as coming next in a sequence on all 1 million of them simultaneously and at every single one of these steps we are making an update to the network for that now the number to watch closely is this number called loss and the loss is a single number that is telling you how well your neural network is performing right now and it is created so that low loss is good so you'll see that the loss is decreasing as we make more updates to the neural
并试图同时改进对这 100 万个标记在序列中下一个出现的预测。在每一个这样的步骤中，我们都在为这个目的更新网络。现在要密切关注的数字是这个叫做“损失”（loss）的数字。损失是一个单一的数字，它告诉你你的神经网络目前的表现如何。设定的规则是损失越低越好。所以你会看到，随着我们对神经网络进行更多的更新，损失在下降。
(36:13) nut which corresponds to making better predictions on the next token in a sequence and so the loss is the number that you are watching as a neural network researcher and you are kind of waiting you're twiddling your thumbs uh you're drinking coffee and you're making sure that this looks good so that with every update your loss is improving and the network is getting better at prediction now here you see that we are processing 1 million tokens per update each update takes about 7 Seconds roughly and here we are going to process
这对应着对序列中的下一个标记做出更好的预测。所以作为一名神经网络研究人员，损失是你要关注的数字。你在等待，无所事事，喝着咖啡，确保一切看起来正常，这样每次更新时，你的损失都在改善，网络的预测能力也在提高。现在你可以看到，我们每次更新处理 100 万个标记，每次更新大约需要 7 秒，我们总共要处理……
(36:44) a total of 32,000 steps of optimization so 32,000 steps with 1 million tokens each is about 33 billion tokens that we are going to process and we're currently only about 420 step 20 out of 32,000 so we are still only a bit more than 1% done because I've only been running this for 10 or 15 minutes or something like that now every 20 steps I have configured this optimization to do inference so what you're seeing here is the model is predicting the next token in a sequence and so you sort of start it randomly and then you continue
总共 32000 步优化。所以 32000 步，每步处理 100 万个标记，大约要处理 330 亿个标记。我们目前才进行到 32000 步中的 420 步左右，所以我们只完成了 1% 多一点，因为我才运行了大约 10 到 15 分钟。现在，我设置每 20 步进行一次推理。所以你在这里看到的是模型在预测序列中的下一个标记。你基本上是随机开始，然后继续……
(37:20) plugging in the tokens so we're running this inference step and this is the model sort of predicting the next token in the sequence and every time you see something appear that's a new token um so let's just look at this and you can see that this is not yet very coherent and keep in mind that this is only 1% of the way through training and so the model is not yet very good at predicting the next token in the sequence so what comes out is actually kind of a little bit of gibberish right but it still has a little bit of like
输入标记。我们在运行这个推理步骤，这是模型在预测序列中的下一个标记。每次你看到有东西出现，那就是一个

新的标记。我们来看看这个，你可以看到它还不是很连贯。要记住，这才训练了 1%，所以模型还不太擅长预测序列中的下一个标记。所以输出的内容实际上有点像胡言乱语，对吧？但它还是有一点……
(37:49) local coherence so since she is mine it's a part of the information should discuss my father great companions Gordon showed me sitting over at and Etc so I know it doesn't look very good but let's actually scroll up and see what it looked like when I started the optimization so all the way here at step one so after 20 steps of optimization you see that what we're getting here is looks completely random and of course that's because the model has only had 20 updates to its parameters and so it's giving you random text because it's a
局部的连贯性，比如 “since she is mine it's a part of the information should discuss my father great companions Gordon showed me sitting over at and Etc”。我知道这看起来不太好，但我们向上滚动看看优化开始时是什么样子。在第一步这里，经过 20 步优化后，你可以看到我们得到的内容看起来完全是随机的。当然，这是因为模型的参数只更新了 20 次，所以它给你的是随机文本，因为它是一个……
(38:24) random Network and so you can see that at least in comparison to this model is starting to do much better and indeed if we waited the entire 32,000 steps the model will have improved the point that it's actually uh generating fairly coherent English uh and the tokens stream correctly um and uh they they kind of make up English a a lot better um so this has to run for about a day or two more now and so uh at this stage we just make sure that the loss is decreasing everything is looking good um and we just have to wait
这是一个随机的网络。所以你可以看到，相比之下，这个模型至少已经开始有了明显进步。实际上，如果我们完成全部 32000 步训练，模型将会提升到能够生成相当连贯的英语，标记流也会正确无误，组合起来的英语也会更加通顺。目前这个训练还需要再运行大概一两天，在这个阶段，我们只需确保损失值持续下降，一切看起来正常，然后耐心等待。
(38:58) and now um let me turn now to the um story of the computation that's required because of course I'm not running this optimization on my laptop that would be way too expensive uh because we have to run this neural network and we have to improve it and we have we need all this data and so on so you can't run this too well on your computer uh because the network is just too large uh so all of this is running on the computer that is out there in the cloud and I want to basically address the compute side of the store of training these models and
现在，我来讲讲训练所需的计算资源相关的情况。当然，我不会在笔记本电脑上运行这个优化过程，因为那代价太大了。我们要运行神经网络，要对其进行优化，还需要大量数据等等，电脑根本无法很好地处理这些任务，因为网络规模太大了。所以，所有这些操作都是在云端的计算机上进行的。我主要想讲讲训练这些模型时计算资源方面的情况，以及……
(39:29) what that looks like so let's take a look okay so the computer that I'm running this optimization on is this 8X h100 node so there are eight h100s in a single node or a single computer now I am renting this computer and it is somewhere in the cloud I'm not sure where it is physically actually the place I like to rent from is called Lambda but there are many other companies who provide this service so when you scroll down you can see that uh they have some on demand pricing for um sort of computers that have these uh
这是怎样的一种情况，我们来看看。我用来运行这个优化的计算机是 8X h100 节点，也就是说，在单个节点或单台计算机中有 8 个 h100。我租了这台计算机，它在云端的某个地方，我其实并不清楚它的实际物理位置。我常租用的平台叫 Lambda，不过还有很多其他公司也提供这种服务。向下滚动页面，你会看到，对于配备这些……
(40:02) h100s which are gpus and I'm going to show you what they look like in a second but on demand 8times Nvidia h100 uh GPU this machine comes for $3 per GPU per hour for example so you can rent these and then you get a machine in a cloud and you can uh go in and you can train these models and these uh gpus they look like this so this is one h100 GPU uh this is kind of what it looks like and you slot this into your computer and gpus are this uh perfect fit for training your networks because they are very computationally expensive but they
h100（它们是GPU）的计算机，他们有按需定价服务。比如，按需租用8块英伟达h100 GPU的机器，每块GPU每小时收费3美元。你可以租用这些设备，然后在云端获得一台机器，进而在上面训练模型。这些GPU长这样，这就是一个h100 GPU，大概就是这个样子。你把它插到计算机里，GPU非常适合训练网络，因为虽然训练计算量很大，但……
(40:39) display a lot of parallelism in the computation so you can have many independent workers kind of um working all at the same time in solving uh the matrix multiplication that's under the hood of training these neural networks so this is just one of these h100s but actually you would put them you would put multiple of them together so you could stack eight of them into a single node and then you can stack multiple nodes into an entire data center or an entire system so when we look at a data center can't spell when we look at a
它们在计算过程中能展现出很强的并行性。所以，你可以让很多独立的计算单元同时工作，去处理训练神经网络背后的矩阵乘法运算。这只是其中一个h100，实际上，你会把多个h100组合在一起，比如把8个h100堆叠到一个节点

中，然后再把多个节点组合成一个完整的数据中心或整个系统。当我们看一个数据中心（拼写有误，应为“center”），当我们观察……

(41:15) data center we start to see things that look like this right so we have one GPU goes to eight gpus goes to a single system goes to many systems and so these are the bigger data centers and there of course would be much much more expensive um and what's happening is that all the big tech companies really desire these gpus so they can train all these language models because they are so powerful and that has is fundamentally what has driven the stock price of Nvidia to be $3.

一个数据中心时，我们会看到类似这样的情况，对吧？从一个 GPU 扩展到 8 个 GPU，再到单个系统，然后扩展到多个系统，这些就是更大的数据中心，它们的成本当然要高得多。实际情况是，所有大型科技公司都非常渴望得到这些 GPU，这样他们就能训练各种语言模型，因为 GPU 功能强大。从根本上说，这就是推动英伟达股价达到3……

(41:42) 4 trillion today as an example and why Nvidia has kind of exploded so this is the Gold Rush the Gold Rush is getting the gpus getting enough of them so they can all collaborate to perform this optimization and they're what are they all doing they're all collaborating to predict the next token on a data set like the fine web data set this is the computational workflow that that basically is extremely expensive the more gpus you have the more tokens you can try to predict and improve on and you're going to process this data set faster and you can iterate

以万亿美元为例，这也是英伟达股价飙升的原因。这就像是一场淘金热，大家都在竞相获取 GPU，尽可能多地得到它们，这样就能协同进行这种优化。它们都在做什么呢？它们都在协同工作，在像 Fine Web 数据集这样的数据集上预测下一个标记。这就是计算工作流程，成本非常高昂。你拥有的 GPU 越多，就能尝试预测和改进更多的标记，处理数据集的速度也就越快，还能更快速地进行迭代。

(42:15) faster and get a bigger Network and train a bigger Network and so on so this is what all those machines are look like are uh are doing and this is why all of this is such a big deal and for example this is a article from like about a month ago or so this is why it's a big deal that for example Elon Musk is getting 100,000 gpus uh in a single Data Center and all of these gpus are extremely expensive are going to take a ton of power and all of them are just trying to predict the next token in the sequence and improve

更快地构建和训练更大规模的网络等等。这就是那些机器正在做的事情，这也是为什么这一切如此重要。例如，大约一个月前有一篇文章提到，埃隆・马斯克在单个数据中心配备 10 万个 GPU，这可不是小事。这些 GPU 极其昂贵，耗电量巨大，它们都在努力预测序列中的下一个标记，进而改进……

(42:46) the network uh by doing so and uh get probably a lot more coherent text than what we're seeing here a lot faster okay so unfortunately I do not have a couple 10 or hundred million of dollars to spend on training a really big model like this but luckily we can turn to some big tech companies who train these models routinely and release some of them once they are done training so they've spent a huge amount of compute to train this network and they release the network at the end of the optimization so it's very useful because

网络，通过这样做，可能会比我们现在看到的更快地生成更加连贯的文本。很遗憾，我没有几千万甚至上亿美元来训练这样一个大型模型，但幸运的是，我们可以借助一些大型科技公司的成果。这些公司经常训练这些模型，训练完成后会发布其中一部分。他们投入了大量的计算资源来训练网络，并在优化结束后发布网络，这非常有用，因为……

(43:16) they've done a lot of compute for that so there are many companies who train these models routinely but actually not many of them release uh these what's called base models so the model that comes out at the end here is is what's called a base model what is a base model it's a token simulator right it's an internet text token simulator and so that is not by itself useful yet because what we want is what's called an assistant we want to ask questions and have it respond to answers these models won't do that they just uh create sort

他们为此进行了大量的计算工作。有很多公司经常训练这些模型，但实际上，很少有公司会发布所谓的基础模型。最终训练出来的模型就是基础模型。什么是基础模型呢？它是一个标记模拟器，本质上是一个互联网文本标记模拟器。就其本身而言，它还没什么实际用处，因为我们想要的是一个助手，我们希望能向它提问并得到回答，而这些模型做不到，它们只是…… 生成一些类似……

(43:46) of remixes of the internet they dream internet pages so the base models are not very often released because they're kind of just only a step one of a few other steps that we still need to take to get in system however a few releases have been made so as an example the gbt2 model released the 1.6 billion sorry 1.

互联网内容的重新组合，像是在“幻想”互联网页面。所以基础模型并不经常发布，因为它只是我们构建可用系统所需的多个步骤中的第一步。不过，还是有一些基础模型被发布了。例如，GPT - 2 模型发布了 16 亿（抱歉，应为 15 亿）……

(44:08) 5 billion model back in 2019 and this gpt2 model is a base model now what is a model release what does it look like to release these models so this is the gpt2 repository on GitHub well you need two things basically to release model number one we need the um python code usually that describes the sequence of operations in detail that they make in their model so um if you remember back this Transformer the sequence of steps that are taken here in this neural network is what is being described by this code so this code is sort of implementing the

2019 年发布的 15 亿参数模型，这个 GPT - 2 模型就是一个基础模型。现在，什么是模型发布呢？发布这些模型是什么样的呢？这是 GitHub 上的 GPT - 2 代码库。基本上，发布模型需要两样东西。第一，我们通常需要 Python 代码，它详细描述了模型中的操作步骤。还记得这个 Transformer 吗？神经网络中所采取的一系列步骤就是由这段代码描述的，所以这段代码在某种程度上实现了……

(44:47) what's called forward pass of this neural network so we need the specific details of exactly how they wired up that neural network so this is just computer code and it's usually just a couple hundred lines of code it's not it's not that crazy and uh this is all fairly understandable and usually fairly standard what's not standard are the parameters that's where the actual value is what are the parameters of this neural network because there's 1.

神经网络的所谓前向传播过程。我们需要确切了解他们是如何构建这个神经网络的具体细节。这只是计算机代码，通常只有几百行，没那么复杂，而且都比较容易理解，通常也很规范。不规范的是参数，参数才是真正有价值的部分。这个神经网络的参数是什么呢？因为有 1……

(45:11) 6 billion of them and we need the correct setting or a really good setting and so that's why in addition to this source code they release the parameters which in this case is roughly 1.5 billion parameters and these are just numbers so it's one single list of 1.5 billion numbers the precise and good setting of all the knobs such that the tokens come out well so uh you need those two things to get a base model release now gpt2 was released but that's actually a fairly old model as I mentioned so actually the model we're

60 亿个参数，我们需要正确的或者非常好的参数设置。这就是为什么除了源代码，他们还会发布参数，在这个例子中，大约有 15 亿个参数。这些只是数字，是一个包含 15 亿个数字的列表，是所有参数的精确且理想的设置，能让标记生成得更好。所以，要发布一个基础模型，你需要这两样东西。现在 GPT - 2 已经发布了，但正如我提到的，它实际上是一个相当老的模型。实际上，我们接下来要看的模型是……

(45:46) going to turn to is called llama 3 and that's the one that I would like to show you next so llama 3 so gpt2 again was 1.6 billion parameters trained on 100 billion tokens Lama 3 is a much bigger model and much more modern model it is released and trained by meta and it is a 45 billion parameter model trained on 15 trillion tokens in very much the same way just much much bigger um and meta has also made a release of llama 3 and that was part of this paper so with this paper that goes into a lot of detail the biggest base model

Llama 3，这就是我接下来要给你展示的。Llama 3，GPT - 2 有 16 亿个参数，在 1000 亿个标记上进行训练。Llama 3 是一个大得多且更现代的模型，由 Meta 发布并训练，它有 450 亿个参数，在 15 万亿个标记上进行训练，训练方式大致相同，只是规模大得多。Meta 也发布了 Llama 3，这在相关论文中有介绍。在这篇详细的论文中，最大的基础模型……

(46:23) that they released is the Lama 3.1 4.5 405 billion parameter model so this is the base model and then in addition to the base model you see here foreshadowing for later sections of the video they also released the instruct model and the instruct means that this is an assistant you can ask it questions and it will give you answers we still have yet to cover that part later for now let's just look at this base model this token simulator and let's play with it and try to think about you know what is this thing and how does it work and

是 Llama 3.1，有 4050 亿个参数的模型。这就是基础模型，除了基础模型，你可以看到这里为视频后面的内容埋下了伏笔，他们还发布了指令模型。“instruct” 意味着这是一个助手，你可以向它提问，它会给出回答。我们稍后再讲这部分内容，现在我们先看看这个基础模型，这个标记模拟器，来玩玩它，思考一下它到底是什么，是如何工作的，以及……

(46:54) um what do we get at the end of this optimization if you let this run Until the End uh for a very big neural network on a lot of data so my favorite place to interact with the base models is this um company called hyperbolic which is basically serving the base model of the 405b Llama 3.

如果让这个在大量数据上训练的大型神经网络一直运行下去，优化结束后我们能得到什么。我最喜欢与基础模型进行交互的平台是一家叫 Hyperbolic 的公司，它提供 4050 亿参数的 Llama 3 基础模型服务。

(47:13) 1 so when you go to the website and I think you may have to register and so on make sure that in the models make sure that you are using llama 3.1 405 billion base it must be the base model and then here let's say the max tokens is how many tokens we're going to be gener rating so let's just decrease this to be a bit less just so we don't waste compute we just want the next 128 tokens and leave the other stuff alone I'm not going to go into the full detail here um now fundamentally what's going to happen here is identical to what happens here during inference

所以当你访问这个网站时，我想你可能需要注册之类的。在模型选项中，确保你选择的是 Llama 3.1 4050 亿参数的基础模型，必须是基础模型。然后这里，假设 “max tokens” 表示我们要生成的标记数量，我们把它调小一点，这样就不会浪费计算资源，我们只想要接下来的 128 个标记，其他设置保持不变。这里我就不详细讲了。从根本上说，这里发生的事情和推理过程是一样的。

(47:43) for us so this is just going to continue the token sequence of whatever you prefix you're going to give it so I want to first show you that this model here is not yet an assistant so you can for example ask it what is 2 plus 2 it's not going to tell you oh it's four uh what else can I help you with it's not going to do that because what is 2 plus 2 is going to be tokenized and then those tokens just act as a prefix and then what the model is going to do now is just going to get the probability for the next token and it's just a glorified

对于我们来说，它只是会根据你输入的前缀继续生成标记序列。我想先让你看看这个模型还不是一个助手。比如，你问它 2 加 2 等于多少，它不会回答你 “哦，等于 4，还有什么我可以帮忙的吗”。因为 “2 加 2 等于多少” 会被标记化，然后这些标记只是作为前缀，接下来模型要做的就是获取下一个标记的概率，它其实就是一个高级版的……

(48:12) autocomplete it's a very very expensive autocomplete of what comes next um depending on the statistics of what it saw in its training documents which are basically web pages so let's just uh hit enter to see what tokens it comes up with as a continuation okay so here it kind of actually answered the question and started to go off into some philosophical territory uh let's try it again so let me copy and paste and let's try again from scratch what is 2 plus two so okay so it just goes off again so notice one more thing that I want to

自动补全功能，只是一个极其昂贵的自动补全工具，根据它在训练文档（基本上是网页）中看到的统计信息来预测接下来的内容。那么我们现在按回车键，看看它会接着生成什么标记。好的，它在这里实际上回答了问题，然后又开始转到一些哲学领域的内容。我们再试一次，我把问题复制粘贴一下，重新开始尝试。2 加 2 等于多少？好的，它又开始偏离主题了。注意，我还想强调一点……

(48:51) stress is that the system uh I think every time you put it in it just kind of starts from scratch so it doesn't uh the system here is stochastic so for the same prefix of tokens we're always getting a different answer and the reason for that is that we get this probity distribution and we sample from it and we always get different samples and we sort of always go into a different territory uh afterwards so here in this case um I don't know what this is let's try one more time so it just continues on so it's just doing the stuff that it's saw on

这个系统，我觉得每次输入内容，它几乎都是从头开始。这个系统是随机的，所以对于相同的标记前缀，我们每次得到的答案都不一样。原因是我们根据概率分布进行采样，每次采样的结果都不同，之后就会进入不同的方向。在这种情况下，我不知道这是什么，我们再试一次。它还是继续这样。它只是在重复它在互联网上看到的内容……

(49:26) the internet right um and it's just kind of like regurgitating those uh statistical patterns so first things it's not an assistant yet it's a token autocomplete and second it is a stochastic system now the crucial thing is that even though this model is not yet by itself very useful for a lot of applications just yet um it is still very useful because in the task of predicting the next token in the sequence the model has learned a lot about the world and it has stored all that knowledge in the parameters of the network so remember that our text

对吧，它只是在重复那些统计模式。首先，它还不是一个助手，只是一个标记自动补全工具；其次，它是一个随机系统。关键在于，尽管这个模型目前本身对很多应用来说还不是特别有用，但它仍然有很大价值。因为在预测序列中下一个标记的任务过程中，模型学到了很多关于世界的知识，并将这些知识存储在网络的参数中。记住，我们的文本……

(50:04) looked like this right internet web pages and now all of this is sort of compressed in the weights of the network so you can think of um these 405 billion parameters is a kind of compression of the internet you can think of the 45 billion parameters is kind of like a zip file uh but it's not a loss less compression it's a loss C compression we're kind of like left with kind of a gal of the internet and we can generate from it right now we can elicit some of this knowledge by prompting the base model uh accordingly so for example

就像这样，是互联网网页内容，而现在所有这些都被压缩到了网络的权重中。你可以把这 4050 亿个参数看作是对互联网的一种压缩，450 亿个参数有点像一个压缩文件，但这不是无损压缩，而是有损压缩。我们从互联网中提取了一部分内容，现在可以通过相应地向基础模型提问来引出其中的一些知识。例如……

(50:38) here's a prompt that might work to elicit some of that knowledge that's hiding in the parameters here's my top 10 list of the top landmarks to see in the pairs um and I'm doing it this way because I'm trying to Prime the model to now continue this list so let's see if that works when I press enter okay so you see that it started a list and it's now kind of giving me some of those landmarks and now notice that it's trying to give a lot of information here now you might not be able to actually fully trust some of the information here
这里有一个提示，可能会引出隐藏在参数中的一些知识："我列出了巴黎十大必看地标"。我这样做是为了引导模型接着列出这个清单。我们按回车键看看是否有效。好的，你看它开始列清单了，现在它给了我一些地标。注意，它在这里试图给出很多信息。现在，你可能不能完全相信这里的一些信息……
(51:11) remember that this is all just a recollection of some of the internet documents and so the things that occur very frequently in the internet data are probably more likely to be remembered correctly compared to things that happen very infrequently so you can't fully trust some of the things that and some of the information that is here because it's all just a vague recollection of Internet documents because the information is not stored explicitly in any of the parameters it's all just the recollection that said we did get
要记住，这些都只是对一些互联网文档的回忆。所以与很少出现的内容相比，在互联网数据中频繁出现的内容更有可能被正确记住。所以你不能完全相信这里的一些内容和信息，因为这些都只是对互联网文档的模糊回忆。信息并没有明确存储在任何参数中，只是一种回忆。尽管如此，我们确实得到了……
(51:39) something that is probably approximately correct and I don't actually have the expertise to verify that this is roughly correct but you see that we've elicited a lot of the knowledge of the model and this knowledge is not precise and exact this knowledge is vague and probabilistic and statistical and the kinds of things that occur often are the kinds of things that are more likely to be remembered um in the model now I want to show you a few more examples of this model's Behavior the first thing I want to show you is this example I went to
一些可能大致正确的内容。我实际上没有专业知识来验证这些是否大致正确，但你可以看到我们引出了模型的很多知识。这些知识并不精确，是模糊的、概率性的和统计性的。在模型中，经常出现的内容更有可能被记住。现在我想给你展示这个模型行为的更多例子。我想给你展示的第一个例子是，我访问了……
(52:08) the Wikipedia page for zebra and let me just copy paste the first uh even one sentence here and let me put it here now when I click enter what kind of uh completion are we going to get so let me just hit enter there are three living species etc etc what the model is producing here is an exact regurgitation of this Wikipedia entry it is reciting this Wikipedia entry purely from memory and this memory is stored in its parameters and so it is possible that at some point in these 512 tokens the model will uh stray away from the Wikipedia entry but
斑马的维基百科页面，我把这里的第一句话复制粘贴过来，然后放在这里。现在我点击回车键，看看会得到什么样的补全内容。我按回车键，它显示"有三种现存物种等等"。模型在这里输出的内容完全是对这个维基百科条目的重复，它纯粹是凭记忆背诵这个维基百科条目，而这些记忆存储在它的参数中。所以在这 512 个标记的某个地方，模型可能会偏离维基百科条目，但……
(52:46) you can see that it has huge chunks of it memorized here uh let me see for example if this sentence occurs by now okay so this so we're still on track let me check here okay we're still on track it will eventually uh stray away okay so this thing is just recited to a very large extent it will eventually deviate uh because it won't be able to remember exactly now the reason that this happens is because these models can be extremely good at memorization and usually this is not what you want in the final model and
你可以看到它记住了很大一部分内容。我看看，比如到现在这个句子是否出现了。好的，还在正轨上。我再检查一下，好的，仍然正确。但它最终会偏离。它在很大程度上只是在背诵，最终会出现偏差，因为它不可能完全准确地记住。这种情况发生的原因是，这些模型的记忆能力可能非常强，但通常这不是最终模型所期望的效果，而且……
(53:20) this is something called regurgitation and it's usually undesirable to site uh things uh directly uh that you have trained on now the reason that this happens actually is because for a lot of documents like for example Wikipedia when these documents are deemed to be of very high quality as a source like for example Wikipedia it is very often uh the case that when you train the model you will preferentially sample from those sources so basically the model has probably done a few epochs on this data meaning that it has seen this web page
这被称为"背诵"现象，直接引用训练数据中的内容通常不是我们想要的。这种情况发生的实际原因是，对于很多文档，比如维基百科，当这些文档被认为是高质量的数据源时，在训练模型时，通常会优先从这些来源采样数据。所以基本上，模型可能在这些数据上训练了几个轮次，这意味着它看过这个网页……
(53:51) like maybe probably 10 times or so and it's a bit like you like when you read some kind of a text many many times say you read something a 100 times uh then you'll be able to recite it and it's very similar for this model if it sees something way too often it's

going to be able to recite it later from memory except these models can be a lot more efficient um like per presentation than human so probably it's only seen this Wikipedia entry 10 times but basically it has remembered this article exactly in its parameters okay the next thing I
大概 10 次左右。这有点像你反复阅读一篇文章，比如说你读了 100 次，然后你就能背诵它了。对这个模型来说也是如此，如果它频繁看到某样东西，之后就能凭记忆背诵出来。只不过这些模型在记忆方面可能比人类更高效。可能它只看过这个维基百科条目 10 次，但基本上它已经把这篇文章准确地记在了参数中。好的，接下来我……
(54:18) want to show you is something that the model has definitely not seen during its training so for example if we go to the paper uh and then we navigate to the pre - training data we'll see here that uh the data set has a knowledge cut off until the end of 2023 so it will not have seen documents after this point and certainly it has not seen anything about the 2024 election and how it turned out now if we Prime the model with the tokens from the future it will continue the token sequence and it will just take its best guess according to the
想给你展示一些模型在训练过程中肯定没见过的内容。例如，如果我们查看这篇论文，然后找到预训练数据部分，我们会发现这里的数据集知识截止到 2023 年底，所以它没见过这个时间点之后的文档，当然也没见过关于 2024 年选举及其结果的任何内容。现在，如果我们用未来的标记来引导模型，它会继续生成标记序列，并根据……
(54:52) knowledge that it has in its own parameters so let's take a look at what that could look like so the Republican Party kit Trump okay president of the United States from 2017 and let's see what it says after this point so for example the model will have to guess at the running mate and who it's against Etc so let's hit enter so here thingss that Mike Pence was the running mate instead of JD Vance and the ticket was against Hillary Clinton and Tim Kane so this is kind of a interesting parallel universe potentially of what could have happened
它自身参数中的知识进行最佳猜测。我们来看看这会是什么样的。“共和党候选人特朗普，2017 年起担任美国总统”，我们看看之后它会说什么。例如，模型必须猜测竞选伙伴以及对手等等。我们按回车键。这里显示迈克・彭斯是竞选伙伴，而不是 JD・万斯，竞选对手是希拉里・克林顿和蒂姆・凯恩。这有点像是一个有趣的平行宇宙，展示了可能发生的情况……
(55:26) happened according to the LM let's get a different sample so the identical prompt and let's resample so here the running mate was Ronda santis and they ran against Joe Biden and Camala Harris so this is again a different parallel universe so the model will take educated guesses and it will continue the token sequence based on this knowledge um and it will just kind of like all of what we're seeing here is what's called hallucination the model is just taking its best guess uh in a probalistic manner the next thing I
根据语言模型的推测。我们再取一个不同的样本，用相同的提示重新采样。这里显示竞选伙伴是罗恩・德桑蒂斯，他们的对手是乔・拜登和卡玛拉・哈里斯。这又是一个不同的平行宇宙。所以模型会进行有根据的猜测，并根据这些知识继续生成标记序列。我们在这里看到的所有内容都被称为 “幻觉”，模型只是以概率方式进行最佳猜测。接下来我……
(55:57) would like to show you is that even though this is a base model and not yet an assistant model it can still be utilized in Practical applications if you are clever with your prompt design so here's something that we would call a few shot prompt so what it is here is that I have 10 words or 10 pairs and each pair is a word of English column and then a the translation in Korean and we have 10 of them and what the model does here is at the end we have teacher column and then here's where we're going to do a completion of say just five tokens and
想给你展示的是，即使这是一个基础模型，还不是助手模型，但如果你巧妙设计提示，它仍然可以用于实际应用。这里有一个我们称之为 “少样本提示” 的例子。我有 10 个单词或 10 对单词，每对中一个是英文单词，另一个是对应的韩语翻译，共 10 对。模型在这里要做的是，在最后有一个 “teacher” 列，我们在这里只完成 5 个标记的生成。
(56:32) these models have what we call in context learning abilities and what that's referring to is that as it is reading this context it is learning sort of in place that there's some kind of a algorithmic pattern going on in my data and it knows to continue that pattern and this is called kind of like Inc context learning so it takes on the role of a translator and when we hit uh completion we see that the teacher translation is Sim which is correct um and so this is how you can build apps by being clever with your prompting even though we still
这些模型具有我们所说的上下文学习能力，意思是当它读取这个上下文时，它会就地学习，发现我的数据中存在某种算法模式，并知道继续这个模式，这就是所谓的上下文学习。所以它在这里扮演了翻译的角色。当我们点击完成时，我们看到 “teacher” 的翻译是 “Sim”，这是正确的。所以，即使我们现在只有基础模型，通过巧妙设计提示，也可以这样构建应用程序。
(57:07) just have a base model for now and it relies on what we call this um uh in context

learning ability and it is done by constructing what's called a few shot prompt okay and finally I want to show you that there is a clever way to actually instantiate a whole language model assistant just by prompting and the trick to it is that we're structure a prompt to look like a web page that is a conversation between a helpful AI assistant and a human and then the model will continue that conversation so actually to write the prompt I turned to
它依赖于我们所说的上下文学习能力，通过构建少样本提示来实现。好的，最后我想给你展示一种巧妙的方法，仅通过提示就能实例化一个完整的语言模型助手。技巧在于，我们构建一个提示，让它看起来像一个网页，展示一个有帮助的人工智能助手和人类之间的对话，然后模型会接着这个对话继续进行。实际上，为了写这个提示，我求助于……
(57:39) chat gbt itself which is kind of meta but I told it I want to create an llm assistant but all I have is the base model so can you please write my um uh prompt and this is what it came up with which is actually quite good so here's a conversation between an AI assistant and a human the AI assistant is knowledgeable helpful capable of answering wide variety of questions Etc and then here it's not enough to just give it a sort of description it works much better if you create this fot prompt so here's a few terms of human assistant human
ChatGPT 本身，这有点元编程的意思。我告诉它我想创建一个大语言模型助手，但我只有基础模型，所以请它帮我写提示。这是它给出的内容，实际上相当不错。这里是一段人工智能助手和人类的对话：人工智能助手知识渊博、乐于助人，能够回答各种问题等等。然后，仅仅给出这样的描述是不够的，如果创建下面这个少样本提示，效果会好得多。这里有几段人类和助手之间的对话……
(58:13) assistant and we have uh you know a few turns of conversation and then here at the end is we're going to be putting the actual query that we like so let me copy paste this into the base model prompt and now let me do human column and this is where we put our actual prompt why is the sky blue and uh let's uh run assistant the sky appears blue due to the phenomenon called R lights scattering etc etc so you see that the base model is just continuing the sequence but because the sequence looks like this conversation it takes on that
助手之间的对话。然后在最后，我们要放入我们真正的查询内容。我把这个复制粘贴到基础模型的提示中，现在我在“human”列中，这里是我们输入实际提示的地方。“为什么天空是蓝色的”，然后我们运行助手，它回答“天空看起来是蓝色的是由于瑞利散射现象等等”。你可以看到，基础模型只是在继续生成序列，但因为这个序列看起来像这段对话，它就承担了……
(58:50) role but it is a little subtle because here it just uh you know it ends the assistant and then just you know hallucinate Ates the next question by the human Etc so it'll just continue going on and on uh but you can see that we have sort of accomplished the task and if you just took this "why is the sky blue" and if we just refresh this and put it here then of course we don't expect this to work with a base model right we're just going to who knows what we're going to get okay we're just going to get more questions okay so this is one way to
这个角色，但这里有点微妙，因为它在助手回答结束后，又会臆想出人类提出的下一个问题等等，就这样一直继续下去。但你可以看到，我们在某种程度上完成了任务。如果你只是单独拿出“为什么天空是蓝色的”这个问题，刷新后直接输入到基础模型中，当然，我们不能指望基础模型能很好地回答，对吧？我们都不知道会得到什么结果，可能只会得到更多莫名其妙的内容。所以，这是一种……
(59:19) create an assistant even though you may only have a base model okay so this is the kind of brief summary of the things we talked about over the last few minutes now let me zoom out here and this is kind of like what we've talked about so far we wish to train LM assistants like chpt we've discussed the first stage of that which is the pre-training stage and we saw that really what it comes down to is we take Internet documents we break them up into these tokens these atoms of little text chunks and then we predict token
即使只有基础模型也能创建一个助手的方法。好的，这是对我们过去几分钟所讨论内容的简要总结。现在让我梳理一下，这就是我们目前讨论的内容。我们希望训练像 ChatGPT 这样的大语言模型助手，我们已经讨论了第一个阶段，即预训练阶段。我们发现，这个阶段归根结底就是获取互联网文档，将它们分解成这些标记，也就是小文本块的“原子”，然后预测标记……
(59:51) sequences using neural networks the output of this entire stage is this base model it is the setting of The parameters of this network and this base model is basically an internet document simulator on the token level so it can just uh it can generate token sequences that have the same kind of like statistics as Internet documents and we saw that we can use it in some applications but we actually need to do better we want an assistant we want to be able to ask questions and we want the model to give us answers and so we need
序列。整个这个阶段的输出就是基础模型，它是网络参数的设置。这个基础模型基本上是一个在标记层面的互联网

文档模拟器，所以它能够生成与互联网文档具有相似统计特征的标记序列。我们已经看到，它可以在一些应用中使用，但实际上我们需要做得更好。我们想要一个助手，希望能够向它提问并得到回答，所以我们需要……
(1:00:21) to now go into the second stage which is called the post-training stage so we take our base model our internet document simulator and hand it off to post training so we're now going to discuss a few ways to do what's called post training of these models these stages in post training are going to be computationally much less expensive most of the computational work all of the massive data centers um and all of the sort of heavy compute and millions of dollars are the pre-training stage but now we go into the slightly cheaper but
进入第二个阶段，即后训练阶段。我们将基础模型，也就是我们的互联网文档模拟器，交给后训练环节。现在我们要讨论一些对这些模型进行后训练的方法。后训练阶段的计算成本要低得多，大部分计算工作，包括大规模数据中心的运算、大量的计算资源投入和数百万美元的花费，都集中在预训练阶段。但现在我们进入的这个阶段，虽然成本稍低，但……
(1:00:53) still extremely important stage called post trining where we turn this llm model into an assistant so let's take a look at how we can get our model to not sample internet documents but to give answers to questions so in other words what we want to do is we want to start thinking about conversations and these are conversations that can be multi-turn so so uh there can be multiple turns and they are in the simplest case a conversation between a human and an assistant and so for example we can imagine the conversation could look
仍然非常重要，这个阶段叫做后训练阶段，在这个阶段我们将大语言模型转变为助手。那么让我们看看如何让模型不再只是采样互联网文档内容，而是能够回答问题。换句话说，我们要开始思考对话，而且这些对话可以是多轮的。在最简单的情况下，就是人类和助手之间的对话。例如，我们可以想象这样的对话场景……
(1:01:22) something like this when a human says "what is 2 plus2" the assistant should re respond with something like "2 plus 2 is 4" when a human follows up and says "what if it was star instead of a plus" assistant could respond with something like this um and similar here this is another example showing that the assistant could also have some kind of a personality here uh that it's kind of like nice and then here in the third example I'm showing that when a human is asking for something that we uh don't wish to help with we can produce what's called
是这样的：当人类问 “2 加 2 等于多少” 时，助手应该回答 “2 加 2 等于 4”；当人类接着问 “如果把加号换成星号呢”，助手可以给出类似这样的回答。这里还有一个类似的例子，展示了助手也可以有某种个性，比如很友好。在第三个例子中，我展示了当人类询问一些我们不想提供帮助的内容时，我们可以给出所谓的……
(1:01:49) refusal we can say that we cannot help with that so in other words what we want to do now is we want to think through how in a system should interact with the human and we want to program the assistant and Its Behavior in these conversations now because this is neural networks we're not going to be programming these explicitly in code we're not going to be able to program the assistant in that way because this is neural networks everything is done through neural network training on data sets and so because of that we are going
拒绝回答，我们可以说无法提供帮助。换句话说，我们现在要思考系统应该如何与人类交互，并且要对助手在这些对话中的行为进行编程。由于这是神经网络，我们不会用代码显式地进行编程，没办法以那种方式对助手进行编程。因为在神经网络中，一切都是通过在数据集上训练神经网络来完成的。因此……
(1:02:17) to be implicitly programming the assistant by creating data sets of conversations so these are three independent examples of conversations in a data dat set an actual data set and I'm going to show you examples will be much larger it could have hundreds of thousands of conversations that are multi- turn very long Etc and would cover a diverse breath of topics but here I'm only showing three examples but the way this works basically is uh a assistant is being programmed by example and where is this data coming from like
我们要通过创建对话数据集来隐式地对助手进行编程。这里展示的是数据集中三个独立的对话示例，实际的数据集要大得多，可能包含数十万轮、非常长的对话，涵盖各种不同的主题。但我这里只展示了三个例子。其基本原理是，助手通过示例进行编程。那么这些数据来自哪里呢？比如……
(1:02:47) 2 * 2al 4 same as 2 plus 2 Etc where does that come from this comes from Human labelers so we will basically give human labelers some conversational context and we will ask them to um basically give the ideal assistant response in this situation and a human will write out the ideal response for an assistant in any situation and then we're going to get the model to basically train on this and to imitate those kinds of responses so the way this works then is we are going to take our base model which we produced in the preing stage
“2 * 2 和 2 加 2 都等于 4” 等等，这些数据来自哪里呢？它们来自人类标注员。我们基本上会给人类标注员

一些对话上下文，让他们给出在这种情况下助手的理想回答。人类会写出在任何情况下助手的理想回答，然后我们让模型在这些数据上进行训练，模仿这些回答。所以其工作方式是，我们将使用在预训练阶段生成的基础模型……

(1:03:20) and this base model was trained on internet documents we're now going to take that data set of internet documents and we're gonna throw it out and we're going to substitute a new data set and that's going to be a data set of conversations and we're going to continue training the model on these conversations on this new data set of conversations and what happens is that the model will very rapidly adjust and will sort of like learn the statistics of how this assistant responds to human queries and then later during inference

这个基础模型是在互联网文档上训练的。现在我们要舍弃那个互联网文档数据集，用一个新的数据集来代替，这个新数据集就是对话数据集。我们将在这个新的对话数据集上继续训练模型。这样一来，模型会快速调整，并且会学习助手如何回应人类查询的统计规律。然后在之后的推理过程中……

(1:03:48) we'll be able to basically um Prime the assistant and get the response and it will be imitating what the humans will human labelers would do in that situation if that makes sense so we're going to see examples of that and this is going to become bit more concrete I also wanted to mention that this post-training stage we're going to basically just continue training the model but um the pre-training stage can in practice take roughly three months of training on many thousands of computers the post-training stage will typically

我们基本上就可以激发助手给出回应，而且它会模仿人类标注员在那种情况下的做法，如果这能说得通的话。我们会看到相关的例子，这样会更具体一些。我还想提到，在后训练阶段，我们基本上只是继续训练模型，但是预训练阶段实际上可能需要在数千台计算机上花费大约三个月的时间进行训练，而后训练阶段通常……

(1:04:17) be much shorter like 3 hours for example um and that's because the data set of conversations that we're going to create here manually is much much smaller than the data set of text on the internet and so this training will be very short but fundamentally we're just going to take our base model we're going to continue training using the exact same algorithm the exact same everything except we're swapping out the data set for conversations so the questions now are what are these conversations how do we represent them how do we get the model

会短得多，比如可能只需要 3 个小时。这是因为我们在这里手动创建的对话数据集比互联网上的文本数据集小得多，所以这个训练会非常短。但从根本上说，我们只是使用基础模型，使用完全相同的算法继续训练，除了将数据集换成对话数据集之外，其他一切都不变。所以现在的问题是，这些对话是什么样的？我们如何表示它们？我们如何让模型……

(1:04:47) to see conversations instead of just raw text and then what are the outcomes of um this kind of training and what do you get in a certain like psychological sense uh when we talk about the model so let's turn to those questions now so let's start by talking about the tokenization of conversations everything in these models has to be turned into tokens because everything is just about token sequences so how do we turn conversations into token sequences is the question and so for that we need to design some kind of ending coding and uh

看到对话而不只是原始文本呢？这种训练的结果是什么？当我们谈论模型时，从某种心理学意义上讲，我们能得到什么呢？现在让我们来探讨这些问题。我们先从对话的标记化开始讲起。在这些模型中，所有内容都必须转换为标记，因为一切都与标记序列有关。所以问题是，我们如何将对话转换为标记序列呢？为此，我们需要设计某种编码方式，并且……

(1:05:17) this is kind of similar to maybe if you're familiar you don't have to be with for example the TCP/IP packet in um on the internet there are precise rules and protocols for how you represent information how everything is structured together so that you have all this kind of data laid out in a way that is written out on a paper and that everyone can agree on and so it's the same thing now happening in llms we need some kind of data structures and we need to have some rules around how these data structures like conversations get

这有点类似于（如果你熟悉的话，不熟悉也没关系）互联网上的 TCP/IP 数据包，对于如何表示信息、所有内容如何组织在一起，都有精确的规则和协议，这样所有这些数据就以一种写在纸上、大家都能理解的方式呈现出来。大语言模型中现在也是如此，我们需要某种数据结构，并且需要围绕对话这样的数据结构如何……

(1:05:44) encoded and decoded to and from tokens and so I want to show you now how I would recreate uh this conversation in the token space so if you go to Tech tokenizer I can take that conversation and this is how it is represented in uh for the language model so here we have we are iterating a user and an assistant in this two- turn conversation and what you're seeing here is it looks ugly but it's actually relatively simple the way it gets turned into a token sequence here at the end is a little bit complicated but at the end

编码和解码为标记制定一些规则。现在我想给你展示我如何在标记空间中重新创建这段对话。如果你使用 Tech tokenizer，我可以输入这段对话，这就是它在语言模型中的表示方式。在这里，我们在这个两轮对话中交替出现用户和助手的内容。你现在看到的可能看起来很复杂，但实际上相对简单。最后将其转换为标记序列的方式有点复杂，但最终……

(1:06:18) this conversation between a user and assistant ends up being 49 tokens it is a one-dimensional sequence of 49 tokens and these are the tokens okay and all the different llms will have a slightly different format or protocols and it's a little bit of a wild west right now but for example GPT 40 does it in the following way you have this special token called <|im_start|> and this is short for <|imaginary_monologue_start|> then you have to specify um I don't actually know why it's called that to be honest then you have to specify

这段用户和助手之间的对话最终变成了 49 个标记，这是一个由 49 个标记组成的一维序列，这些就是标记。不同的大语言模型会有稍微不同的格式或协议，现在这方面有点像“西部荒野”（比较混乱）。例如，GPT 40 是这样做的：有一个特殊的标记 <|im_start|>，它是 <|imaginary_monologue_start|> 的缩写，然后你必须指定…… 说实话，我也不知道为什么这么叫。然后你必须指定……

(1:06:53) whose turn it is so for example "user" which is a token 4 28 then you have internal monologue separator and then it's the exact question so the tokens of the question and then you have to close it so <|im_end|> the end of the imaginary monologue so basically the question from a user of "what is 2 plus two" ends up being the token sequence of these tokens and now the important thing to mention here is that <|im_start|> this is not text right <|im_start|> is a special token that gets added it's a new token and um this token has

轮到谁发言，比如 “user”，它对应的标记是 428，然后是内部独白分隔符，接着是确切的问题，也就是问题的标记，然后你必须结束它，即 <|im_end|>，表示虚构独白的结束。所以基本上，用户的问题 “2 加 2 等于多少” 最终变成了这些标记组成的标记序列。这里要重点提到的是，<|im_start|> 这个不是文本，对吧？<|im_start|> 是一个添加的特殊标记，是一个新标记，而且这个标记……

(1:07:31) never been trained on so far it is a new token that we create in a post-training stage and we introduce and so these special tokens like <|im_sep|> <|im_start|> Etc are introduced and interspersed with text so that they sort of um get the model to learn that hey this is a the start of a turn for who is it start of the turn for the start of the turn is for the user and then this is what the user says and then the user ends and then it's a new start of a turn and it is by the assistant and then what does the assistant say well these are the

到目前为止从未在训练中出现过，它是我们在后训练阶段创建并引入的新标记。所以像 <|im_sep|>、<|im_start|> 等这些特殊标记被引入并穿插在文本中，这样可以让模型学习到，嘿，这是一轮对话的开始，这一轮是谁的呢？这一轮是用户的开始，然后这是用户说的话，然后用户说完了，接着是新的一轮开始，这一轮是助手的，然后助手说什么呢？这些就是……

(1:08:03) tokens of what the assistant says Etc and so this conversation is not turned into the sequence of tokens the specific details here are not actually that important all I'm trying to show you in concrete terms is that our conversations which we think of as kind of like a structured object end up being turned via some encoding into one-dimensional sequences of tokens and so because this is one dimensional sequence of tokens we can apply all the stuff that we applied before now it's just a sequence of tokens and now we can train a language

助手说的话的标记等等。所以这段对话被转换为标记序列，这里的具体细节实际上并不是那么重要。我只是想具体地向你展示，我们认为是结构化对象的对话，最终通过某种编码被转换为一维标记序列。因为这是一个一维标记序列，我们可以应用之前用过的所有方法。现在它只是一个标记序列，我们可以在上面训练语言……

(1:08:33) model on it and so we're just predicting the next token in a sequence uh just like before and um we can represent and train on conversations and then what does it look like at test time during inference so say we've trained a model and we've trained a model on these kinds of data sets of conversations and now we want to inference so during inference what does this look like when you're on on chash apt well you come to chash apt and you have say like a dialogue with it and the way this works is basically um say that this was already

在这个序列上训练模型，所以我们就像之前一样预测序列中的下一个标记。我们可以表示对话并在对话上进行训练，那么在测试时，也就是推理阶段是什么样的呢？假设我们训练了一个模型，并且是在这类对话数据集上进行训练的，现在我们要进行推理。当你使用 ChatGPT 进行推理时是什么样的呢？嗯，你来到 ChatGPT，和它进行对话，其工作方式基本上是这样的：假设已经有了……

(1:09:06) filled in so like "what is 2 plus 2" "2 plus 2 is four" and now you issue "what if it was times <|im_end|>" and what basically ends up happening um on the servers of open AI or something like that is they put in <|im_start|>assistant<|im_sep|> and this is where they end it right here so they construct this context and now they start sampling from the

model so it's at this stage that they will go to the model and say okay what is a good for sequence what is a good first token what is a good second token what is a good third token and this is where the LM
一些输入，比如 “2 加 2 等于多少”“2 加 2 等于 4”，现在你输入 “如果是乘号呢 <|im_end|>”，基本上在 OpenAI 的服务器或类似的地方发生的是，他们输入 <|im_start|>assistant<|im_sep|>，就在这里结束输入。所以他们构建了这个上下文，现在他们开始从模型中采样。就是在这个阶段，他们会向模型询问，比如，什么样的序列是合适的，第一个标记选什么好，第二个标记选什么好，第三个标记选什么好，就在这个时候，语言模型……
(1:09:38) takes over and creates a response like for example response that looks something like this but it doesn't have to be identical to this but it will have the flavor of this if this kind of a conversation was in the data set so um that's roughly how the protocol Works although the details of this protocol are not important so again my goal is that just to show you that everything ends up being just a one-dimensional token sequence so we can apply everything we've already seen but we're now training on conversations and we're
开始发挥作用并生成一个回答，例如生成一个类似这样的回答，但不一定完全一样，如果数据集中有类似这样的对话，那么生成的回答会有相似的风格。这就是大致的流程，虽然这个流程的细节并不重要。我再次强调，我的目的只是向你展示，最终一切都变成了一维标记序列。所以我们可以应用我们之前了解到的所有知识，只不过现在我们是在对话上进行训练，而且我们……
(1:10:09) now uh basically generating conversations as well okay so now I would like to turn to what these data sets look like in practice the first paper that I would like to show you and the first effort in this direction is this paper from openai in 2022 and this paper was called instruct GPT or the technique that they developed and this was the first time that opena has kind of talked about how you can take language models and fine-tune them on conversations and so this paper has a number of details that I would like to
现在基本上也在生成对话。好的，现在我想讲讲这些数据集在实际中的情况。我想给你展示的第一篇论文，也是在这个方向上的首次尝试，是 OpenAI 在 2022 年发表的一篇论文，叫做《指令微调 GPT》（instruct GPT），或者说是他们开发的一种技术。这是 OpenAI 首次谈到如何利用语言模型并在对话上对其进行微调。这篇论文中有很多细节，我想……
(1:10:36) take you through so the first stop I would like to make is in section 3.4 where they talk about the human contractors that they hired uh in this case from upwork or through scale AI to uh construct these conversations and so there are human labelers involved whose job it is professionally to create these conversations and these labelers are asked to come up with prompts and then they are asked to also complete the ideal assistant responses and so these are the kinds of prompts that people came up with so these are human labelers
带你了解一下。我首先想讲的是 3.4 节的内容，他们在这部分谈到了他们雇佣的人力外包人员，这些人是通过 Upwork 平台或者 Scale AI 公司找来构建这些对话的。这里涉及到人类标注员，他们的工作是专业创建这些对话。这些标注员被要求提出问题，然后还要给出理想的助手回答。下面是人们提出的一些问题示例，这些都是人类标注员提出的……
(1:11:07) so list five ideas for how to regain enthusiasm for my career what are the top 10 science fiction books I should read next and there's many different types of uh kind of prompts here so translate this sentence from uh to Spanish Etc and so there's many things here that people came up with they first come up with the prompt and then they also uh answer that prompt and they give the ideal assistant response now how do they know what is the ideal assistant response that they should write for these prompts so when we scroll down a
比如，列出五条重新找回职业热情的方法；我接下来应该读的十佳科幻小说有哪些。这里有很多不同类型的问题，比如把这句话从…… 翻译成西班牙语等等。人们想出了很多这样的问题，他们先提出问题，然后回答这些问题，并给出理想的助手回答。那么他们怎么知道针对这些问题应该写什么样的理想助手回答呢？当我们往下滚动……
(1:11:36) little bit further we see that here we have this excerpt of labeling instructions uh that are given to the human labelers so the company that is developing the language model like for example open AI writes up labeling instructions for how the humans should create ideal responses and so here for example is an excerpt uh of these kinds of labeling instruction instructions on High level you're asking people to be helpful truthful and harmless and you can pause the video if you'd like to see more here but on a high level basically
再往下一点，我们看到这里有一段给人类标注员的标注说明节选。开发语言模型的公司，比如 OpenAI，会编写标注说明，指导人类如何创建理想的回答。例如，这里有一段这类标注说明的节选，从总体上来说，就是要求人们要乐于助人、真实可靠且无害。如果你想了解更多，可以暂停视频。总体而言，基本上……
(1:12:04) just just answer try to be helpful try to be truthful and don't answer questions that we don't want um kind of the system to handle uh later in chat gbt and so roughly

speaking the company comes up with the labeling instructions usually they are not this short usually there are hundreds of pages and people have to study them professionally and then they write out the ideal assistant responses uh following those labeling instructions so this is a very human heavy process as it was described in this paper now the data set for instruct

就是回答问题时要尽量提供帮助、保证真实，对于那些我们不希望系统在之后（比如在 ChatGPT 中）处理的问题不要回答。大致来说，公司会制定标注说明，通常这些说明不会这么简短，往往有几百页，人们需要专业学习这些说明，然后按照这些说明写出理想的助手回答。正如论文中所描述的，这是一个人力密集型的过程。现在，用于指令微调 GPT（instruct GPT）的数据集……

(1:12:35) GPT was never actually released by openi but we do have some open- Source um reproductions that were're trying to follow this kind of a setup and collect their own data so one that I'm familiar with for example is the effort of open Assistant from a while back and this is just one of I think many examples but I just want to show you an example so here's so these were people on the internet that were asked to basically create these conversations similar to what um open I did with human labelers and so here's an entry of a person who

OpenAI 实际上从未发布过，但我们有一些开源的复制品，它们试图遵循类似的设置并收集自己的数据。比如，我比较熟悉的一个例子是之前的 Open Assistant 项目。我认为这只是众多例子中的一个，但我想给你展示一下。这里，互联网上的人们被要求创建类似于 OpenAI 让人类标注员创建的对话。下面是一个人的输入示例……

(1:13:05) came up with this "BR can you write a short introduction to the relevance of the term 'manop' uh in economics please use examples Etc" and then the same person or potentially a different person will write up the response so here's the assistant response to this and so then the same person or different person will actually write out this ideal response and then this is an example of maybe how the conversation could continue "explain it to a dog" and then you can try to come up with a slightly a simpler explanation or

这个人提出 “BR，你能写一篇关于‘manop’这个术语在经济学中的相关性的简短介绍吗？请举例说明等等”，然后同一个人或者可能是不同的人会写出回答。下面是助手对这个问题的回答，然后同一个人或不同的人会写出理想的回答。下面是对话可能继续的一个例子，“用给狗狗解释的方式来讲讲”，然后你可以尝试给出一个更简单一点的解释，或者……

(1:13:37) something like that now this then becomes the label and we end up training on this so what happens during training is that um of course we're not going to have a full coverage of all the possible questions that um the model will encounter at test time during inference we can't possibly cover all the possible prompts that people are going to be asking in the future but if we have a like a data set of a few of these examples then the model during training will start to take on this Persona of this helpful truthful harmless assistant

类似的内容。现在这个就成为了标注，我们最终在这个数据上进行训练。在训练过程中会发生什么呢？当然，我们不可能涵盖模型在测试（推理）阶段可能遇到的所有问题，我们无法覆盖人们将来可能提出的所有问题。但是如果我们有一个包含一些这样示例的数据集，那么模型在训练过程中就会开始呈现出这个乐于助人、真实可靠且无害的助手的角色特点。

(1:14:12) and it's all programmed by example and so these are all examples of behavior and if you have conversations of these example behaviors and you have enough of them like 100,000 and you train on it the model sort of starts to understand the statistical pattern and it kind of takes on this personality of this assistant now it's possible that when you get the exact same question like this at test time it's possible that the answer will be recited as exactly what was in the training set but more likely than that is that the model will kind of

这一切都是通过示例进行编程的，所以这些都是行为示例。如果你有足够多这样的对话示例，比如 10 万个，并在这些数据上进行训练，模型就会开始理解其中的统计模式，并在某种程度上呈现出这个助手的个性。现在，在测试时如果你遇到完全相同的问题，答案有可能会和训练集中的完全一样，但更有可能的情况是，模型会……

(1:14:43) like do something of a similar Vibe um and we will understand that this is the kind of answer that you want um so that's what we're doing we're programming the system um by example and the system adopts statistically this Persona of this helpful truthful harmless assistant which is kind of like reflected in the labeling instructions that the company creates now I want to show you that the state-of-the-art has kind of advanced in the last 2 or 3 years uh since the instr GPT paper so in particular it's not very common for

做出类似风格的回答，而我们会理解这就是我们想要的那种答案。这就是我们正在做的事情，我们通过示例对系统进行编程，系统从统计意义上呈现出这个乐于助人、真实可靠且无害的助手的角色特点，这在一定程度上反映在公司制定的标注说明中。现在我想告诉你，自指令微调 GPT 的论文发表后的两三年里，技术水平有了一定的进步。特别是现在，不太常见的情况是……

(1:15:15) humans to be doing all the heavy lifting just by themselves anymore and that's because we now have language models and these language models are helping us create these data sets and conversations so it is very rare that the people will like literally just write out the response from scratch it is a lot more likely that they will use an existing llm to basically like uh come up with an answer and then they will edit it or things like that so there's many different ways in which now llms have started to kind of permeate this
人类再完全靠自己完成所有繁重的工作了，因为我们现在有了语言模型，这些语言模型正在帮助我们创建这些数据集和对话。现在人们很少会完全从头开始写出回答，更有可能的是，他们会使用现有的大语言模型来生成一个答案，然后再进行编辑或类似的操作。所以现在大语言模型已经开始以多种不同的方式渗透到这个……
(1:15:40) posttraining Set uh stack and llms are basically used pervasively to help create these massive data sets of conversations so I don't want to show like Ultra chat is one um such example of like a more modern data set of conversations it is to a very large extent synthetic but uh I believe there's some human involvement I could be wrong with that usually there will be a little bit of human but there will be a huge amount of synthetic help um and this is all kind of like uh constructed in different ways and Ultra chat is just
后训练环节中，大语言模型被广泛用于帮助创建这些大规模的对话数据集。我不想举例太多，UltraChat 就是一个更现代的对话数据集的例子，它在很大程度上是合成的，但我认为其中也有一些人类的参与，我可能说得不对，通常会有少量人力参与，但会有大量的合成数据辅助。这些数据集的构建方式各不相同，UltraChat 只是……
(1:16:10) one example of many sft data sets that currently exist and the only thing I want to show you is that uh these data sets have now millions of conversations uh these conversations are mostly synthetic but they're probably edited to some extent by humans and they span a huge diversity of sort of um uh areas and so on so these are fairly extensive artifacts by now and there's all these like sft mixtures as they're called so you have a mixture of like lots of different types and sources and it's partially synthetic partially
目前众多监督微调（SFT）数据集的一个例子。我想向你展示的只是，这些数据集现在包含数百万个对话，这些对话大多是合成的，但可能在一定程度上经过了人类编辑，它们涵盖了各种各样的领域等等。所以到现在，这些数据集已经是相当广泛的成果了，而且有各种所谓的 SFT 混合数据集。你会看到很多不同类型和来源的混合，部分是合成的，部分是……
(1:16:41) human and it's kind of like um gone in that direction since uh but roughly speaking we still have sft data sets they're made up of conversations we're training on them um just like we did before and uh I guess like the last thing to note is that I want to dispel a little bit of the magic of talking to an AI like when you go to chat GPT and you give it a question and then you hit enter uh what is coming back is kind of like statistically aligned with what's happening in the training set and these training sets I mean they really just
人工生成的，从那以后就朝着这个方向发展了。但大致来说，我们仍然有监督微调数据集，它们由对话组成，我们像以前一样在这些数据集上进行训练。我想最后要指出的是，我想破除一些与人工智能对话的神秘感。比如当你使用 ChatGPT，向它提出一个问题然后点击回车，返回的内容在某种程度上是与训练集中的情况统计相关的。而这些训练集，我的意思是，它们实际上只是……
(1:17:17) have a seed in humans following labeling instructions so what are you actually talking to in chat GPT or how should you think about it well it's not coming from some magical AI like roughly speaking it's coming from something that is statistically imitating human labelers which comes from labeling instructions written by these companies and so you're kind of imitating this uh you're kind of getting um it's almost as if you're asking human labeler and imagine that the answer that is given to you uh from chbt is some kind of a simulation of a
基于人类遵循标注说明生成的数据。那么你在 ChatGPT 中实际上是在和什么对话呢？你应该如何看待它呢？其实它并不是来自某个神奇的人工智能，大致来说，它来自于从统计上模仿人类标注员的东西，而这些标注员的标注是基于公司编写的标注说明。所以你有点像是在模仿这个过程，你得到的回答，几乎就像是在问人类标注员，想象一下，ChatGPT 给你的回答是某种对……
(1:17:48) human labeler uh and it's kind of like asking what would a human labeler say in this kind of a conversation and uh it's not just like this human labeler is not just like a random person from the internet because these companies actually hire experts so for example when you are asking questions about code and so on the human labelers that would be in um involved in creation of these conversation data sets they will usually be usually be educated expert people and you're kind of like asking a question of like a simulation
人类标注员提问，就好像在问在这样的对话中人类标注员会说什么。而且这个人类标注员并非来自互联网上的普通人，因为这些公司实际上聘请的是专家。例如，当你询问有关代码等方面的问题时，参与创建这些对话数据集的人

类标注员通常是受过良好教育的专业人士。你就像是在向一个模拟……
(1:18:17) of those people if that makes sense so you're not talking to a magical AI you're talking to an average labeler this average labeler is probably fairly highly skilled but you're talking to kind of like an instantaneous simulation of that kind of a person that would be hired uh in the construction of these data sets so let me give you one more specific example before we move on for example when I go to chpt and I say "recommend the top five landmarks who see in Paris" and then I hit enter uh okay here we go okay when I hit enter
这些人的模拟体提问，如果这样说能讲得通的话。所以你并不是在和神奇的人工智能对话，你是在和一个普通标注员对话。这个普通标注员可能技能相当高超，但你是在和一种对参与构建这些数据集的人的即时模拟对话。在继续之前，我再给你举一个具体的例子。比如，当我在 ChatGPT 中输入“推荐巴黎的五大必看地标”然后回车，好的，我们看看。当我回车后……
(1:18:52) what's coming out here how do I think about it well it's not some kind of a magical AI that has gone out and researched all the landmarks and then ranked them using its infinite intelligence Etc what I'm getting is a statistical simulation of a labeler that was hired by open AI you can think about it roughly in that way and so if this specific um question is in the posttraining data set somewhere at open aai then I'm very likely to see an answer that is probably very very similar to what that human labeler would
出现的内容是什么呢？我该如何理解它呢？嗯，它并不是某种神奇的人工智能，出去研究了所有地标，然后凭借其无穷的智慧对它们进行排名等等。我得到的是对 OpenAI 聘请的标注员的统计模拟结果。你可以大致这样理解。所以，如果这个具体的问题在 OpenAI 的后训练数据集中，那么我很可能看到一个与那个人类标注员给出的答案非常相似的回答……
(1:19:24) have put down for those five landmarks how does the human labeler come up with this well they go off and they go on the internet and they kind of do their own little research for 20 minutes and they just come up with a list right now so if they come up with this list and this is in the data set I'm probably very likely to see what they submitted as the correct answer from the assistant now if this specific query is not part of the post training data set then what I'm getting here is a little bit more emergent uh
对于这五大地标。人类标注员是怎么想出这个答案的呢？嗯，他们会去网上，花 20 分钟左右做一些自己的小研究，然后列出一个清单。所以，如果他们列出了这个清单并且它在数据集中，那么我很可能看到助手给出的答案和他们提交的正确答案一样。现在，如果这个特定的查询不在后训练数据集中，那么我得到的答案就更具涌现性……
(1:19:51) because uh the model kind of understands the statistically um the kinds of landmarks that are in this training set are usually the prominent landmarks the landmarks that people usually want to see the kinds of landmarks that are usually uh very often talked about on the internet and remember that the model already has a ton of Knowledge from its pre-training on the internet so it's probably seen a ton of conversations about Paris about landmarks about the kinds of things that people like to see and so it's the
因为模型在一定程度上从统计学角度理解，训练集中的地标通常是著名的地标，是人们通常想要参观的地标，是那些在互联网上经常被提及的地标。要记住，模型在互联网上的预训练过程中已经积累了大量知识，所以它可能看过大量关于巴黎、关于地标、关于人们喜欢参观的事物的对话。所以……
(1:20:17) pre-training knowledge that has then combined with the postering data set that results in this kind of an imitation um so that's uh that's roughly how you can kind of think about what's happening behind the scenes here in in this statistical sense okay now I want to turn to the topic of llm psychology as I like to call it which is what are sort of the emergent cognitive effects of the training pipeline that we have for these models so in particular the first one I want to talk to is of course hallucinations so you might be familiar
预训练知识与后训练数据集相结合，产生了这种模仿效果。所以，大致来说，从统计学意义上，你可以这样理解幕后发生的事情。好的，现在我想谈谈我喜欢称之为“大语言模型心理学”的话题，也就是这些模型的训练流程会产生哪些涌现的认知效应。我特别想先谈谈的当然是幻觉问题。你可能对……
(1:20:50) with model hallucinations it's when llms make stuff up they just totally fabricate information Etc and it's a big problem with llm assistants it is a problem that existed to a large extent with early models uh from many years ago and I think the problem has gotten a bit better uh because there are some medications that I'm going to go into in a second for now let's just try to understand where these hallucinations come from so here's a specific example of a few uh of three conversations that you might think you have in your
模型幻觉有所了解，它指的是大语言模型编造内容、完全虚构信息等情况。这是大语言模型助手存在的一个大问题，多年前的早期模型在很大程度上就存在这个问题。我认为这个问题已经有所改善，因为有一些解决方法，我马上会讲到。现在，我们先试着理解这些幻觉是从哪里来的。这里有一个具体例子，是三个你可能认为会出现在训练

集中的对话……

(1:21:17) training set and um these are pretty reasonable conversations that you could imagine being in the training set so like for example "who is Cruz" well "Tom Cruz is an famous actor American actor and producer Etc" "who is John baraso" this turns out to be a us senetor for example "who is genis Khan" well "genis Khan was blah blah blah" and so this is what your conversations could look like at training time now the problem with this is that when the human is writing the correct answer for the assistant in each one of these cases uh the human either

这些对话是比较合理的，你可以想象它们在训练集中。例如，“谁是克鲁兹？”“汤姆·克鲁兹是一位著名的美国演员和制片人等等”；“谁是约翰·巴拉索？” 结果他是一位美国参议员；“谁是吉恩斯·汗？”“吉恩斯·汗是……（此处应补充完整信息）”。所以这些是训练时对话可能的样子。问题在于，当人类为这些情况中的助手编写正确答案时，人类要么……

(1:21:51) like knows who this person is or they research them on the Internet and they come in and they write this response that kind of has this like confident tone of an answer and what happens basically is that at test time when you ask for someone who is this is a totally random name that I totally came up with and I don't think this person exists um as far as I know I just Tred to generate it randomly the problem is when we ask "who is Orson kovats" the problem is that the assistant will not just tell you "oh I don't know" even if the assistant and

知道这个人是谁，要么在网上进行搜索，然后给出回答，而且回答带有一种自信的语气。基本上，在测试时，当你问一个我随便编出来的、我觉得根本不存在的人的名字时，问题就出现了。比如当我们问 “谁是奥森·科瓦茨” 时，问题在于，即使助手（以及语言模型本身）在其特征、激活状态，或者说在它的 “大脑” 中某种程度上知道这个人是它不熟悉的，但助手不会直接告诉你 “哦，我不知道”……

(1:22:20) the language model itself might know inside its features inside its activations inside of its brain sort of it might know that this person is like not someone that um that is that it's familiar with even if some part of the network kind of knows that in some sense the uh saying that "oh I don't know who this is" is is not going to happen because the model statistically imitates is training set in the training set the questions of the form "who is blah" are confidently answered with the correct answer and so it's going to take on the

即使网络的某些部分在某种意义上知道这一点，但它不会说 “哦，我不知道这是谁”。因为模型从统计学上模仿训练集，在训练集中，“谁是……” 这种形式的问题都是用正确答案自信地回答的。所以它会采用……

(1:22:52) style of the answer and it's going to do its best it's going to give you statistically the most likely guess and it's just going to basically make stuff up because these models again we just talked about it is they don't have access to the internet they're not doing research these are statistical token tumblers as I call them uh is just trying to sample the next token in the sequence and it's going to basically make stuff up so let's take a look at what this looks like I have here what's called the inference playground from hugging face

这种回答风格，尽最大努力，从统计学角度给出最可能的猜测，基本上就是在编造内容。因为就像我们刚才说的，这些模型无法访问互联网，它们不会去做研究，我把它们称为统计标记生成器，它们只是试图对序列中的下一个标记进行采样，然后基本上就是在编造内容。我们来看看这是什么样的。我这里有 Hugging Face 的推理游乐场……

(1:23:21) and I am on purpose picking on a model called Falcon 7B which is an old model this is a few years ago now so it's an older model So It suffers from hallucinations and as I mentioned this has improved over time recently but let's say "who is Orson kovats" let's ask Falcon 7B instruct run oh yeah "Orson kovat is an American author and science uh fiction writer" okay this is totally false it's hallucination let's try again these are statistical systems right so we can resample this time "Orson kovat is a fictional character from this 1950s TV

我故意选择了一个叫 Falcon 7B 的模型，这是一个老模型，是几年前的了。所以它存在幻觉问题。正如我提到的，最近这个问题已经有所改善，但我们来试试，比如问 “谁是奥森·科瓦茨”，我们在 Falcon 7B 的指令模式下运行看看。哦，它回答 “奥森·科瓦茨是一位美国作家和科幻小说家”。好的，这完全是错误的，这就是幻觉。我们再试一次，这些是统计系统，对吧？所以我们可以重新采样。这次它说 “奥森·科瓦茨是 20 世纪 50 年代一部电视剧中的虚构角色”……

(1:23:54) show"it's total BS right let's try again he's a former minor league baseball player" okay so basically the model doesn't know and it's given us lots of different answers because it doesn't know it's just kind of like sampling from these probabilities the model starts with the tokens "who is oron kovats assistant" and then it comes in here and it's get it's getting these probabilities and it's just sampling from the probabilities and it just like comes up with stuff and the stuff is actually statistically consistent with the style

这完全是胡扯，对吧？我们再试一次，它说 “他是一名前小联盟棒球运动员”。所以基本上这个模型不知道，它给了我们很多不同的答案，因为它不知道，它只是从这些概率中进行采样。模型从 “who is oron kovats assistant” 这些标记开始，然后它获取这些概率，只是从概率中采样，然后就给出一些内容，而这些内容实际上在统计上与训练集中的回答风格……

(1:24:28) of the answer in its training set and it's just doing that but you and I experiened it as a madeup factual knowledge but keep in mind that uh the model basically doesn't know and it's just imitating the format of the answer and it's not going to go off and look it up uh because it's just imitating again the answer so how can we uh mitigate this because for example when we go to chat apt and I say "who is oron kovats" and I'm now asking the stateoftheart state-of-the-art model from open AI this model will tell you "oh so this model is actually is even

是一致的，它只是在这么做。但你和我会觉得这是编造的事实知识。但要记住，模型基本上是不知道的，它只是在模仿回答的格式，它不会去查找信息，因为它只是在模仿答案。那么我们如何缓解这个问题呢？例如，当我们在 ChatGPT 中问 “谁是奥森・科瓦茨” 时，我现在问的是 OpenAI 最先进的模型，这个模型会告诉你 “哦，实际上这个模型甚至……（此处原文可能有误，句子不完整）”

(1:25:01) smarter because you saw very briefly it said "searching the web uh we're going to cover this later um it's actually trying to do tool use and uh kind of just like came up with some kind of a story but I want to just who or Kovach did not use any tools I don't want it to do web search there's a wellknown historical or public figure named or oron kovats" so this model is not going to make up stuff this model knows that it doesn't know and it tells you that it doesn't appear to be a person that this model knows so

更智能，因为你会短暂看到它显示 “正在搜索网络，我们稍后会讲到这个。它实际上在尝试使用工具，然后编出了某种故事，但我只想说，奥森・科瓦茨，不要使用任何工具，我不想让它进行网络搜索，有一个名为奥森・科瓦茨的著名历史或公众人物”。所以这个模型不会编造内容，这个模型知道自己不知道，并且会告诉你它似乎不认识这个人。所以……

(1:25:36) somehow we sort of improved hallucinations even though they clearly are an issue in older models and it makes totally uh sense why you would be getting these kinds of answers if this is what your training set looks like so how do we fix this okay well clearly we need some examples in our data set that where the correct answer for the assistant is that the model doesn't know about some particular fact but we only need to have those answers be produced in the cases where the model actually doesn't know and so the question is how

不知怎么地，我们在一定程度上改善了幻觉问题，尽管在老模型中这显然是个问题。如果你的训练集是这样的，那么你得到这些答案就完全说得通了。那么我们如何解决这个问题呢？很明显，我们需要在数据集中加入一些例子，在这些例子中，助手的正确答案是模型不知道某些特定事实，但我们只需要在模型真正不知道的情况下给出这样的答案。所以问题是……

(1:26:05) do we know what the model knows or doesn't know well we can empirically probe the model to figure that out so let's take a look at for example how meta uh dealt with hallucinations for the Llama 3 series of models as an example so in this paper that they published from meta we can go into "hallucinations" which they call here "factuality" and they describe the procedure by which they basically interrogate the model to figure out what it knows and doesn't know to figure out sort of like the boundary of its knowledge and then they

我们如何知道模型知道什么或不知道什么呢？我们可以通过实证探究模型来弄清楚。我们以 Meta 处理 Llama 3 系列模型的幻觉问题为例来看一下。在 Meta 发表的这篇论文中，我们可以看到 “hallucinations”（他们在这里称之为 “factuality”，即事实性）这部分内容，他们描述了一种方法，基本上就是通过询问模型来弄清楚它知道什么、不知道什么，找出它的知识边界，然后他们……

(1:26:38) add examples to the training set where for the things where the model doesn't know them the correct answer is that the model doesn't know them which sounds like a very easy thing to do in principle but this roughly fixes the issue and the the reason it fixes the issue is because remember like the model might actually have a pretty good model of its self knowledge inside the network so remember we looked at the network and all these neurons inside the network you might imagine that there's a neuron somewhere in the network that sort of

在训练集中加入例子，对于模型不知道的事情，正确答案就是模型不知道。这在原则上听起来很容易做到，但这大致解决了问题。之所以能解决问题，是因为要记住，模型在网络内部实际上可能对自己的知识有一个很好的 “认知”。还记得我们看过的网络以及网络中的所有神经元吗？你可以想象，在网络的某个地方有一个神经元，它在某种程度上……

(1:27:12) like lights up for when the model is uncertain but the problem is that the activation of that neuron is not currently wired up to the model actually saying in words that it doesn't know so even though the internal of the neural network no because there's

some neurons that represent that the model uh will not surface that it will instead take its best guess so that it sounds confident um just like it sees in a training set so we need to basically interrogate the model and allow it to say "I don't know" in the cases
在模型不确定的时候会被激活，但问题在于，这个神经元的激活目前并没有与模型实际用语言表达“我不知道”联系起来。所以，即使神经网络内部因为某些神经元的存在而知道（自己不确定），但它不会表现出来，而是会做出最好的猜测，让自己听起来很自信，就像它在训练集中看到的那样。所以，我们基本上需要对模型进行询问，并让它在确实不知道的情况下能够说“我不知道”。
(1:27:41) it doesn't know so let me take you through what meta roughly does so basically what they do is here I have an example uh "Dominic kek is uh the featured article today" so I just went there randomly and what they do is basically they take a random document in a training set and they take a paragraph and then they use an llm to construct questions about that paragraph so for example I did that with chat GPT here so I said "here's a paragraph from this document generate three specific factual questions based on this paragraph and give me the questions and
在它不知道的情况下。下面我给你讲讲 Meta 大致是怎么做的。基本上，他们是这样做的：我这里有个例子，“多米尼克・凯克是今日的专题文章主角”，这是我随机选的。他们基本上是从训练集中随机选取一份文档，选其中一段内容，然后用大语言模型针对这段内容提出问题。比如，我在这里用 ChatGPT 来做这件事，我说“这是这份文档中的一段内容，根据这段内容生成三个具体的事实性问题，并给我问题和……
(1:28:18) the answers"and so the llms are already good enough to create and reframe this information so if the information is in the context window um of this llm this actually works pretty well it doesn't have to rely on its memory it's right there in the context window and so it can basically reframe that information with fairly high accuracy so for example can generate questions for us like"for which team did he play"here's the answer"how many cups did he win" Etc and now what we have to do is we have some question and answers and now we want to
答案”。大语言模型已经能够很好地创建和重新组织这些信息。所以，如果信息在这个大语言模型的上下文窗口中，这个方法实际上效果很好，它不必依赖记忆，信息就在上下文窗口里，所以它基本上可以相当准确地重新组织这些信息。例如，它可以为我们生成这样的问题，比如“他效力于哪个球队”，这里给出答案，“他赢了多少个奖杯”等等。现在我们要做的是，我们有了一些问题和答案，然后我们想要……
(1:28:50) interrogate the model so roughly speaking what we'll do is we'll take our questions and we'll go to our model which would be uh say llama uh in meta but let's just interrogate mol 7B here as an example that's another model so does this model know about this answer let's take a look uh so "he played for Buffalo Sabers" right so the model knows and the the way that you can programmatically decide is basically we're going to take this answer from the model and we're going to compare it to the correct answer and again the model model are good enough to
询问模型。大致来说，我们要做的是，拿上我们的问题，找到我们的模型，比如 Meta 的 Llama 模型，不过我们这里以 mol 7B 为例，这是另一个模型。我们来看看这个模型是否知道这个答案。比如“他效力于水牛城军刀队”，对吧？模型知道这个答案。你可以通过编程判断的方式基本上是，我们从模型获取这个答案，然后将其与正确答案进行比较。模型已经足够好，能够……
(1:29:25) do this automatically so there's no humans involved here we can take uh basically the answer from the model and we can use another llm judge to check if that is correct according to this answer and if it is correct that means that the model probably knows so what we're going to do is we're going to do this maybe a few times so okay it knows it's Buffalo Savers let's drag in um Buffalo Sabers let's try one more time Buffalo Sabers so we asked three times about this factual question and the model seems to know so everything is
自动完成这个比较，这里不需要人类参与。我们基本上可以从模型获取答案，然后用另一个大语言模型作为评判来检查这个答案是否正确。如果正确，那就意味着模型可能知道这个答案。所以我们可能会这样做几次。好的，它知道是水牛城军刀队，我们再确认一下，水牛城军刀队，我们再试一次，水牛城军刀队。我们就这个事实性问题问了三次，模型似乎都知道答案，一切都……
(1:29:58) great now let's try the second question "how many Stanley Cups did he win" and again let's interrogate the model about that and the correct answer is two so um here the model claims that he won um four times which is not correct right it doesn't match two so the model doesn't know it's making stuff up let's try again um so here the model again it's kind of like making stuff up right let's Dragon here it says "did he did not even did not win during his career" so obviously the model doesn't know and the way we can programmatically tell again
很好。现在我们试试第二个问题，“他赢了多少个斯坦利杯”，我们再就这个问题询问模型，正确答案是两个。这里模型声称他赢了四次，这是不正确的，对吧？与正确答案不符。所以模型不知道，它在编造内容。我们再试一次。这里模型又在编造内容，对吧？我们再确认一下，它说“他在职业生涯中甚至从未赢过”。很明显模型不知道，我

们再次通过编程判断的方式……

(1:30:43) is we interrogate the model three times and we compare its answers maybe three times five times whatever it is to the correct answer and if the model doesn't know then we know that the model doesn't know this question and then what we do is we take this question we create a new conversation in the training set so we're going to add a new conversation training set and when the question is "how many Stanley Cups did he win" the answer is "I'm sorry I don't know" or "I don't remember" and that's the correct answer for this

是，我们对模型询问三次，将它的答案与正确答案比较，比较三次、五次都可以。如果模型不知道，那么我们就知道模型确实不知道这个问题。然后我们要做的是，拿上这个问题，在训练集中创建一个新的对话。我们要在训练集中添加一个新对话，当问题是 “他赢了多少个斯坦利杯” 时，答案是 “很抱歉，我不知道” 或者 “我不记得了”，对于这个问题来说，这就是正确答案。

(1:31:12) question because we interrogated the model and we saw that that's the case if you do this for many different types of uh questions for many different types of documents you are giving the model an opportunity to in its training set refuse to say based on its knowledge and if you just have a few examples of that in your training set the model will know um and and has the opportunity to learn the association of this knowledge-based refusal to this internal neuron somewhere in its Network that we presume exists and empirically this turns out to

因为我们询问了模型，发现确实如此。如果你针对许多不同类型的问题、不同类型的文档都这样做，你就是在训练集中给模型一个机会，让它基于自己的知识选择不回答。如果在训练集中有一些这样的例子，模型就会知道，并且有机会学习这种基于知识的拒绝回答与我们认为存在于其网络某个地方的内部神经元之间的联系。从经验上看，结果证明……

(1:31:45) be probably the case and it can learn that Association that "hey when this neuron of uncertainty is high then I actually don't know and I'm allowed to say that 'I'm sorry but I don't think I remember this' Etc" and if you have these uh examples in your training set then this is a large mitigation for hallucination and that's roughly speaking why chpt is able to do stuff like this as well so these are kinds of uh mitigations that people have implemented and that have improved the factuality issue over time okay so I've

很可能是这样，它能够学习到这种联系，即 “嘿，当这个表示不确定的神经元活跃程度高时，我实际上不知道，而且我可以说‘很抱歉，我想我不记得这个了’等等”。如果在训练集中有这些例子，那么这就能在很大程度上缓解幻觉问题。大致来说，这就是 ChatGPT 也能做到类似事情的原因。这些就是人们实施的一些缓解措施，随着时间的推移，这些措施改善了事实性问题。好的，我已经……

(1:32:17) described mitigation number one for basically mitigating the hallucinations issue now we can actually do much better than that uh it's instead of just saying that we don't know uh we can introduce an additional mitigation number two to give the llm an opportunity to be factual and actually answer the question now what do you and I do if I was to ask you a factual question and you don't know uh what would you do um in order to answer the question well you could uh go off and do some search and uh use the internet and you could figure out the

描述了缓解幻觉问题的第一种方法。实际上，我们可以做得更好。除了仅仅让模型说不知道，我们可以引入第二种缓解方法，让大语言模型有机会给出符合事实的回答。现在，如果我问你一个事实性问题，而你不知道答案，你会怎么做呢？为了回答这个问题，你可能会去做一些搜索，利用互联网，然后弄清楚……

(1:32:50) answer and then tell me what that answer is and we can do the exact exact same thing with these models so think of the knowledge inside the neural network inside its billions of parameters think of that as kind of a vague recollection of the things that the model has seen during its training during the pre-training stage a long time ago so think of that knowledge in the parameters as something you read a month ago and if you keep reading something then you will remember it and the model remembers that but if it's something

答案，然后告诉我答案是什么。我们可以对这些模型做完全一样的事情。把神经网络中数十亿参数所包含的知识，看作是模型在很久以前的预训练阶段看到的事物的模糊记忆。把参数中的这些知识想象成你一个月前读过的东西，如果你不断阅读某样东西，你就会记住它，模型也是这样记住知识的。但是如果是……

(1:33:19) rare then you probably don't have a really good recollection of that information but what you and I do is we just go and look it up now when you go and look it up what you're doing basically is like you're refreshing your working memory with information and then you're able to sort of like retrieve it talk about it or Etc so we need some equivalent of allowing the model to refresh its memory or its recollection and we can do that by introducing tools uh for the models so the way we are going to approach this is that instead of just

很少见的内容，那么你可能对这些信息没有很好的记忆。但你和我会怎么做呢？我们会去查找信息。现在当你去查

找信息时，你基本上是在用新信息刷新你的工作记忆，然后你就能检索信息、谈论信息等等。所以我们需要找到一种方法，让模型能够刷新它的记忆或回忆，我们可以通过为模型引入工具来实现这一点。我们的方法是，不再只是……

(1:33:46) saying "hey I'm sorry I don't know" we can attempt to use tools so we can create uh a mechanism by which the language model can emit special tokens and these are tokens that we're going to introduce new tokens so for example here I've introduced two tokens and I've introduced a format or a protocol for how the model is allowed to use these tokens so for example instead of answering the question when the model does not instead of just saying "I don't know sorry" the model has the option now to emitting the special token "search_start"

说 “嘿，很抱歉，我不知道”，我们可以尝试使用工具。我们可以创建一种机制，让语言模型能够发出特殊标记，这些是我们要引入的新标记。例如，我在这里引入了两个标记，并且引入了一种格式或协议，规定模型如何使用这些标记。比如，当模型不知道答案时，它不再只是说 “很抱歉，我不知道”，现在模型可以选择发出特殊标记 “search_start”……

(1:34:18) and this is the query that will go to like bing.com in the case of openai or say Google search or something like that so it will emit the query and then it will emit "search_end" and then here what will happen is that the program that is sampling from the model that is running the inference when it sees the special token "search_end" instead of sampling the next token uh in the sequence it will actually pause generating from the model it will go off it will open a session with bing.com and it will paste the search query into Bing and it will then

这个查询会发送到（以 OpenAI 为例）必应（bing.com）或者谷歌搜索之类的搜索引擎。所以它会发出查询内容，然后发出 “search_end”。接下来会发生的是，运行推理、从模型采样的程序，当它看到 “search_end” 这个特殊标记时，不会继续对序列中的下一个标记进行采样，而是会暂停从模型生成内容，它会去打开一个与必应的会话，把搜索查询内容粘贴到必应中，然后……

(1:34:49) um get all the text that is retrieved and it will basically take that text it will maybe represent it again with some other special tokens or something like that and it will take that text and it will copy paste it here into what I Tred to like show with the brackets so all that text kind of comes here and when the text comes here it enters the context window so the model so that text from the web search is now inside the context window that will feed into the neural network and you should think of the context window as kind of like the working memory of the model

获取检索到的所有文本。它基本上会获取这些文本，可能会用一些其他特殊标记重新表示这些文本，然后把这些文本复制粘贴到我用括号示意的地方。所以所有这些文本就到了这里，当文本到达这里时，它进入上下文窗口。对于模型来说，来自网络搜索的文本现在就在上下文窗口中，这个上下文窗口会输入到神经网络中。你可以把上下文窗口看作是模型的工作记忆……

(1:35:20) that data that is in the context window is directly accessible by the model it directly feeds into the neural network so it's not anymore a vague recollection it's data that it it has in the context window and is directly available to that model so now when it's sampling the new uh tokens here afterwards it can reference very easily the data that has been copy pasted in there so that's roughly how these um how these tools use uh tools uh function and so web search is just one of the tools we're going to look at some of the other tools in a bit uh but basically you introduce new tokens you introduce some schema by which the model can utilize these tokens and can call these special functions like web search functions and how do you teach the model how to correctly use these tools like say web search search_start search_end Etc well again you do that through training sets so we need now to have a bunch of data and a bunch of conversations that show the model by

上下文窗口中的数据模型可以直接访问，它直接输入到神经网络中。所以这不再是模糊的记忆，而是模型在上下文窗口中拥有的、可以直接使用的数据。所以现在，当它之后对新的标记进行采样时，它可以很容易地参考粘贴在那里的数据。这就是这些工具的大致工作原理。网络搜索只是其中一种工具，我们一会儿还会看一些其他工具。但基本上，你引入新标记，引入一些模式，让模型能够使用这些标记并调用像网络搜索功能这样的特殊函数。那么如何教会模型正确使用这些工具，比如网络搜索、“search_start”“search_end” 等等呢？同样，你要通过训练集来实现。所以我们现在需要有大量数据和大量对话，通过示例向模型展示……

(1:35:52) example how to use web search so what are the what are the settings where you are using the search um and what does that look like and here's by example how you start a search and the search Etc and uh if you have a few thousand maybe examples of that in your training set the model will actually do a pretty good job of understanding uh how this tool works and it will know how to sort of structure its queries and of course because of the pre-training data set and its understanding of the world it actually kind of understands what a web

如何使用网络搜索。比如在什么情况下使用搜索，搜索是什么样的，这里举例说明如何开始搜索和结束搜索等等。

如果你在训练集中有几千个这样的例子，模型实际上会很好地理解这个工具是如何工作的，它会知道如何组织查询内容。当然，由于预训练数据集以及它对世界的理解，它实际上对网络搜索有一定的理解……
(1:36:21) search is and so it actually kind of has a pretty good native understanding um of what kind of stuff is a good search query um and so it all kind of just like works you just need a little bit of a few examples to show it how to use this new tool and then it can lean on it to retrieve information and uh put it in the context window and that's equivalent to you and I looking something up because once it's in the context it's in the working memory and it's very easy to manipulate and access so that's what we saw a few minutes ago
网络搜索是什么，实际上它对什么样的内容适合作为搜索查询有很好的内在理解。所以这一切基本上都能行得通。你只需要用几个例子向它展示如何使用这个新工具，然后它就可以依靠这个工具检索信息并将其放入上下文窗口中。这就相当于你我去查找信息，因为一旦信息进入上下文，就进入了工作记忆，很容易进行操作和访问。这就是我们几分钟前看到的情况。
(1:36:50) when I was searching on chat GPT for "who is Orson kovats" the chat GPT language model decided that this is some kind of a rare um individual or something like that and instead of giving me an answer from its memory it decided that it will sample a special token that is going to do web search and we saw briefly something flash it was like "using the web tool" or something like that so it briefly said that and then we waited for like two seconds and then it generated this and you see how it's creating references here and so it's
当我在 ChatGPT 上搜索 “谁是奥森・科瓦茨” 时，ChatGPT 语言模型判断这是某种不太常见的人物之类的，它没有从自己的记忆中给出答案，而是决定采样一个特殊标记来进行网络搜索。我们短暂地看到屏幕上闪过一些提示，类似于 “正在使用网络工具” 之类的内容。它短暂地显示了这个提示，然后我们等了大概两秒，接着它生成了内容。你看它在这里引用了很多内容，它……
(1:37:18) citing sources so what happened here is it went off it did a web web search it found these sources and these URLs and the text of these web pages was all stuffed in between here and it's not showing here but it's it's basically stuffed as text in between here and now it sees that text and now it kind of references it and says that "okay it could be these people citation could be those people citation Etc" so that's what happened here and that's what and that's why when I said "who is Orson kovats" I could also say "don't use any tools" and
在引用资料来源。这里发生的事情是，它进行了网络搜索，找到了这些资料来源和网址，这些网页的文本内容都被塞到了这里（虽然没有显示出来，但基本上就是以文本形式放在这里）。现在它看到了这些文本，然后引用这些内容，说 “好的，可能是这些人，引用来源可能是那些人等等”。这就是这里发生的事情，这也是为什么当我说 “谁是奥森・科瓦茨” 时，我也可以说 “不要使用任何工具”，然后……
(1:37:46) then that's enough to um basically convince chat PT to not use tools and just use its memory and its recollection I also went off and I um tried to ask this question of Chachi PT so "how many standing cups did uh Dominic Hasek win" and Chachi P actually decided that it knows the answer and it has the confidence to say that uh he want twice and so it kind of just relied on its memory because presumably it has um it has enough of a kind of confidence in its weights in it parameters and activations that this is uh retrievable just for memory um but
这样基本上就能让 ChatGPT 不使用工具，而只依靠它的记忆和回忆。我还尝试在 ChatGPT 上问了这个问题：“多米尼克・哈塞克赢得了多少个斯坦利杯？”ChatGPT 实际上判断它知道答案，并且很自信地说他赢了两次。所以它基本上是依靠自己的记忆，大概是因为它对自己的权重、参数和激活状态有足够的信心，认为这个信息可以从记忆中检索到。但是……
(1:38:22) you can also conversely use web search to make sure and then for the same query it actually goes off and it searches and then it finds a bunch of sources it finds all this all of this stuff gets copy pasted in there and then it tells us uh to again and sites and it actually says the Wikipedia article which is the source of this information for us as well so that's tools web search the model determines when to search and then uh that's kind of like how these tools uh work and this is an additional kind of mitigation for uh hallucinations and
相反，你也可以使用网络搜索来确认信息。对于同样的查询，它会进行搜索，找到一堆资料来源，把所有这些内容都复制粘贴进去，然后再次告诉我们答案，并给出引用。它实际上提到了维基百科文章，这也是我们获取信息的来源。这就是网络搜索工具，模型决定何时进行搜索，这就是这些工具的大致工作方式。这是对幻觉问题的另一种缓解方法，并且……
(1:38:59) factuality so I want to stress one more time this very important sort of psychology Point knowledge in the parameters of the neural network is a vague recollection the knowledge in the tokens that make up the context window is the working memory and it roughly speaking Works kind of like um it works for us in our brain the stuff we remember

is our parameters uh and the stuff that we just experienced like a few seconds or minutes ago and so on you can imagine that being in our context window and this context window is being

有助于提高内容的事实准确性。所以我想再次强调这个非常重要的心理学观点：神经网络参数中的知识是一种模糊的记忆，构成上下文窗口的标记中的知识是工作记忆。大致来说，这和我们大脑的工作方式有点类似，我们记住的东西就像是参数，而我们刚刚经历的，比如几秒或几分钟前的事情，你可以想象这些在我们的上下文窗口中。这个上下文窗口……

(1:39:35) built up as you have a conscious experience around you so this has a bunch of um implications also for your use of LOLs in practice so for example I can go to chat GPT and I can do something like this I can say "can you Summarize chapter one of Jane Austin's Pride and Prejudice" right and this is a perfectly fine prompt and Chach actually does something relatively reasonable here and but the reason it does that is because Chach has a pretty good recollection of a famous work like Pride and Prejudice it's probably seen a ton

随着你周围有意识的体验而不断积累。这对你在实际中使用大语言模型也有很多影响。例如，我可以在 ChatGPT 中这样做，我可以说 “你能总结一下简・奥斯汀《傲慢与偏见》的第一章吗？” 这是一个完全没问题的提示。ChatGPT 在这里实际上给出了相对合理的回答，但它能这样做的原因是，它对《傲慢与偏见》这样的著名作品有很好的记忆，它可能看过大量……

(1:40:06) of stuff about it there's probably forums about this book it's probably read versions of this book um and it's kind of like remembers because even if you've read this or articles about it you'd kind of have a recollection enough to actually say all this but usually when I actually interact with LMS and I want them to recall specific things it always works better if you just give it to them so I think a much better prompt would be something like this "can you summarize for me chapter one of genos's spr and Prejudice" and then I am

关于这本书的内容，可能有很多关于这本书的论坛，它可能读过这本书的一些版本。它有点像记住了这些内容，因为即使你读过这本书或者关于它的文章，你也会有足够的记忆来讲述这些内容。但通常当我与大语言模型交互，希望它们回忆特定内容时，直接把内容提供给它们效果会更好。所以我认为一个更好的提示可能是这样：“你能为我总结一下简・奥斯汀《傲慢与偏见》的第一章吗？” 然后我……

(1:40:34) attaching it below for your reference and then I do something like a delimeter here and I paste it in and I I found that just copy pasting it from some website that I found here um so copy pasting the chapter one here and I do that because when it's in the context window the model has direct access to it and can exactly it doesn't have to recall it it just has access to it and so this summary is can be expected to be a significantly high quality or higher quality than this summary uh just because it's directly available to the

在下面附上内容供你参考，然后我在这里用一个分隔符，把内容粘贴进去。我是从这里找到的某个网站上复制粘贴的。我这样做是因为当内容在上下文窗口中时，模型可以直接访问它，它不必去回忆，直接就能获取到。所以这样得到的总结可以预期比仅靠模型记忆生成的总结质量要高得多，因为内容直接可用。

(1:41:03) model and I think you and I would work in the same way if you want to it would be you would produce a much better summary if you had reread this chapter before you had to summarize it and that's basically what's happening here or the equivalent of it the next sort of psychological Quirk I'd like to talk about briefly is that of the knowledge of self so what I see very often on the internet is that people do something like this they ask llms something like "what model are you" and "who built you" and um basically this uh question is a

模型。我想你和我在这方面的工作方式是一样的，如果你要总结某内容，在总结之前重新阅读相关章节，你会写出更好的总结。这里基本上就是这种情况。接下来我想简要谈谈另一种心理学上的奇怪现象，即模型的自我认知。我经常在网上看到人们这样做，他们问大语言模型一些类似 “你是什么模型”“谁创建了你” 的问题。基本上，这个问题有点……

(1:41:33) little bit nonsensical and the reason I say that is that as I try to kind of explain with some of the underhood fundamentals this thing is not a person right it doesn't have a persistent existence in any way it sort of boots up processes tokens and shuts off and it does that for every single person it just kind of builds up a context window of conversation and then everything gets deleted and so this this entity is kind of like restarted from scratch every single conversation if that makes sense it has no persistent self it has no

没有意义。我这么说的原因是，正如我试图从一些底层原理来解释的，这个东西不是一个人，对吧？它没有任何持续存在的概念。它启动、处理标记，然后关闭，对每一个与之交互的人都是如此。它只是构建一个对话上下文窗口，然后所有内容都会被删除。所以这个实体在每一次对话中都像是从头开始重启，如果这样说能理解的话。它没有持续的自我，没有……

(1:42:01) sense of self it's a token tumbler and uh it follows the statistical regularities of its training set so it doesn't really make sense to ask it "who are you" "what build you" Etc and by default if you do what I described and just by default and from nowhere you're going to get some pretty random answers so for example let's uh pick on Falcon which is a fairly old model and let's see what it tells us uh so it's evading the question uh "talented engineers and developers here" it says "I was built by open AI based on the gpt3 model" it's totally making stuff

自我意识，它只是一个标记生成器，并且遵循训练集的统计规律。所以问它 “你是谁”“谁创建了你” 等问题并没有实际意义。默认情况下，如果你按照我描述的那样提问，你会得到一些相当随机的答案。比如，我们以 Falcon 这个比较老的模型为例，看看它会告诉我们什么。它在回避问题，说 “这里有才华横溢的工程师和开发者”，还说 “我是由 OpenAI 基于 GPT - 3 模型构建的”，这完全是在编造内容。

(1:42:29) up now the fact that it's built by open AI here I think a lot of people would take this as evidence that this model was somehow trained on open AI data or something like that I don't actually think that that's necessarily true the reason for that is that if you don't explicitly program the model to answer these kinds of questions then what you're going to get is its statistical best guess at the answer and this model had a um sft data mixture of conversations and during the fine-tuning um the model sort of understands as it's training on this

现在它说自己是由 OpenAI 构建的，我想很多人会把这当作这个模型在某种程度上是在 OpenAI 的数据上进行训练的证据之类的。但我实际上并不认为这一定是真的。原因是，如果你没有明确对模型进行编程，让它回答这类问题，那么你得到的只是它从统计角度给出的最佳猜测答案。这个模型有一个包含对话的监督微调（SFT）混合数据集，在微调过程中……

(1:42:57) data that it's taking on this personality of this like helpful assistant and it doesn't know how to it doesn't actually it wasn't told exactly what label to apply to self it just kind of is taking on this uh this uh Persona of a helpful assistant and remember that the pre-training stage took the documents from the entire internet and Chach and open AI are very prominent in these documents and so I think what's actually likely to be happening here is that this is just its hallucinated label for what it is this is its self-identity

模型在训练过程中逐渐形成了这种乐于助人的助手角色。它不知道如何…… 实际上它没有被明确告知该给自己贴上什么样的标签，只是在扮演一个乐于助人的助手角色。要记住，预训练阶段使用的是来自整个互联网的文档，ChatGPT 和 OpenAI 在这些文档中非常突出。所以我认为这里实际可能发生的情况是，这只是它虚构出来的自我标签，这是它的自我认知……

(1:43:27) is that it's chat GPT by open Ai and it's only saying that because there's a ton of data on the internet of um answers like this that are actually coming from open from chasht and So that's its label for what it is now you can override this as a developer if you have a llm model you can actually override it and there are a few ways to do that so for example let me show you there's this MMO model from Allen Ai and um this is one llm it's not a top tier LM or anything like that but I like it because it is fully open source so the

是自己是 OpenAI 的 ChatGPT。它这么说是因为互联网上有大量这样的答案，实际上这些答案来自 ChatGPT。所以这就是它给自己的标签。作为开发者，如果你有一个大语言模型，你可以覆盖这个设定，有几种方法可以做到这一点。比如，我给你展示一下 Allen AI 的 MMO 模型。这是一个大语言模型，它不是顶级的大语言模型，但我喜欢它是因为它是完全开源的。所以……

(1:43:55) paper for Almo and everything else is completely fully open source which is nice um so here we are looking at its sft mixture so this is the data mixture of um the fine tuning so this is the conversations data it right and so the way that they are solving it for Theo model is we see that there's a bunch of stuff in the mixture and there's a total of 1 million conversations here but here we have alot to hardcoded if we go there we see that this is 240 conversations and look at these 240 conversations they're hardcoded "tell me

关于这个模型的论文以及其他所有内容都是完全开源的，这很不错。我们现在看它的监督微调混合数据集，这是微调时使用的数据混合，也就是对话数据。他们为这个模型解决自我认知问题的方式是，我们看到混合数据集中有很多内容，这里总共有 100 万个对话，但这里有很多是硬编码的内容。如果我们查看，会发现有 240 个对话是硬编码的，看看这 240 个对话，它们硬编码了 “告诉我……

(1:44:25) about yourself"says user and then the assistant says"I'm and open language model developed by AI to Allen Institute of artificial intelligence Etc I'm here to help blah blah blah what is your name uh Theo project" so these are all kinds of like cooked up hardcoded questions abouto 2 and the correct answers to give in these cases if you take 240 questions like this or conversations put them into your training set and fine tune with it then the model will actually be expected to parot this stuff later if you don't

关于你自己”，用户这样问，然后助手回答 “我是由艾伦人工智能研究所开发的开源语言模型等等。我在这里提供帮助，等等。你叫什么名字？西奥项目”。所以这些都是为这个模型硬编码的各种问题和相应的正确答案。如果你把 240 个这样的问题或对话放入训练集并进行微调，那么之后模型就会被期望照本宣科地回答这些内容。如果你不……

(1:44:53) give it this then it's probably a Chach by open Ai and um there's one more way to sometimes do this is that basically um in these conversations and you have terms between human and assistant sometimes there's a special message called system message at the very beginning of the conversation so it's not just between human and assistant there's a system and in the system message you can actually hardcode and remind the model that "hey you are a model developed by open Ai and your name is chashi pt40 and you were trained on

如果不这样设置，它可能就会自称是 OpenAI 开发的 ChatGPT。还有一种有时会用到的方法，基本上在这些人类与助手的对话中，在对话的最开始，除了人类和助手之间的交互信息外，还有一种特殊消息，叫做系统消息。也就是说，这里不只是人类和助手在对话，还有系统参与其中。在系统消息中，你可以进行硬编码设置，提醒模型，比如 “嘿，你是 OpenAI 开发的模型，你的名字是 ChatGPT 40，你是在……

(1:45:17) this date and your knowledge cut off is this and basically it kind of like documents the model a little bit and then this is inserted into to your conversations so when you go on chpt you see a blank page but actually the system message is kind of like hidden in there and those tokens are in the context window and so those are the two ways to kind of um program the models to talk about themselves either it's done through uh data like this or it's done through system message and things like that basically invisible tokens that are

这个日期进行训练的，你的知识截止于这个时间”，基本上就是对模型进行一些记录说明，然后将这些内容插入到对话中。所以当你打开 ChatGPT 时，你看到的是一个空白页面，但实际上系统消息就隐藏在那里，这些标记就在上下文窗口中。所以，大致有两种方式可以对模型进行设置，让它们能介绍自己：一种是通过像这样的数据设置，另一种是通过系统消息，也就是那些基本上不可见的标记，它们……

(1:45:44) in the context window and remind the model of its identity but it's all just kind of like cooked up and bolted on in some in some way it's not actually like really deeply there in any real sense as it would before a human I want to now continue to the next section which deals with the computational capabilities or like I should say the native computational capabilities of these models in problem solving scenarios and so in particular we have to be very careful with these models when we construct our examples of conversations

在上下文窗口中提醒模型它的身份。但这一切在某种程度上都像是人为设定并附加上去的，并不像人类对自身身份的认知那样真实深刻。现在我想接着进入下一部分内容，这部分讨论这些模型在解决问题场景中的计算能力，或者更确切地说，是它们的原生计算能力。所以，特别要注意的是，当我们构建对话示例来训练这些模型时，必须非常小心。

(1:46:11) and there's a lot of sharp edges here that are kind of like elucidative is that a word uh they're kind of like interesting to look at when we consider how these models think so um consider the following prompt from a human and supposed that basically that we are building out a conversation to enter into our training set of conversations so we're going to train the model on this we're teaching you how to basically solve simple math problems so the prompt is "Emily buys three apples and two oranges each orange cost $2 the total

这里有很多需要深入探讨的地方，当我们思考这些模型的思维方式时，这些点很有意思。我们来考虑一下人类给出的以下提示，假设我们正在构建一个对话，准备将其放入对话训练集中。我们要用这个对话来训练模型，教它如何解决简单的数学问题。提示内容是：“艾米丽买了 3 个苹果和 2 个橙子，每个橙子 2 美元，总共花费……

(1:46:39) cost is 13 what is the cost of apples" very simple math question now there are two answers here on the left and on the right they are both correct answers they both say that the answer is three which is correct but one of these two is a significant ific anly better answer for the assistant than the other like if I was Data labeler and I was creating one of these one of these would be uh a really terrible answer for the assistant and the other would be okay and so I'd like you to potentially pause the video Even and think through why one of these

13 美元，苹果的单价是多少？” 这是一个非常简单的数学问题。现在这里有两个答案，左边和右边的答案都是正确的，都表明答案是 3，这是正确的。但对于助手来说，这两个答案中有一个明显比另一个更好。如果我是数据标注员，在创建这样的答案时，其中一个对助手来说会是一个非常糟糕的答案，而另一个则还可以。所以我希望你甚至可以暂停视频，思考一下为什么这两个答案中……

(1:47:10) two is significantly better answer uh than the other and um if you use the wrong one your model will actually be uh really bad at math potentially and it would have uh bad outcomes and this is something that you would be careful with in your life labeling

documentations when you are training people uh to create the ideal responses for the assistant okay so the key to this question is to realize and remember that when the models are training and also inferencing they are working in one - dimensional sequence of tokens from
有一个明显更好。如果你用了错误的那个答案，你的模型可能在数学方面表现得非常差，会产生不好的结果。这是你在编写标注文档，训练人们为助手创建理想回答时需要注意的问题。好的，这个问题的关键在于要认识并记住，当模型进行训练和推理时，它们处理的是从左到右的一维标记序列。
(1:47:38) left to right and this is the picture that I often have in my mind I imagine basically the token sequence evolving from left to right and to always produce the next token in a sequence we are feeding all these tokens into the neural network and this neural network then is the probabilities for the next token and sequence right so this picture here is the exact same picture we saw uh before up here and this comes from the web demo that I showed you before right so this is the calculation that basically takes
这是我脑海中常有的一个概念。我大致想象标记序列从左到右展开，为了生成序列中的下一个标记，我们将所有这些标记输入到神经网络中，然后神经网络会给出下一个标记在序列中的概率，对吧？这里的这幅图和我们之前看到的完全一样，它来自我之前给你展示的网络演示，对吧？这就是那个计算过程，基本上就是……
(1:48:06) the input tokens here on the top and uh performs these operations of all these neurons and uh gives you the answer for the probabilities of what comes next now the important thing to realize is that roughly speaking uh there's basically a finite number of layers of computation that happened here so for example this model here has only one two three layers of what's called detention and uh MLP here um maybe um typical modern state-of-the-art Network would have more like say 100 layers or something like that but there's only 100 layers of
将顶部的输入标记进行所有这些神经元的运算，然后给出下一个标记的概率。现在要认识到的重要一点是，大致来说，这里的计算层数是有限的。例如，这个模型这里只有一、二、三层所谓的注意力层（detention，疑为“attention”）和多层感知器（MLP）。也许典型的现代最先进的网络会有大概 100 层左右，但这里从之前的标记序列到计算下一个标记的概率，只有 100 层左右的计算……
(1:48:39) computation or something like that to go from the previous token sequence to the probabilities for the next token and so there's a finite amount of computation that happens here for every single token and you should think of this as a very small amount of computation and this amount of computation is almost roughly fixed uh for every single token in this sequence um the that's not actually fully true because the more tokens you feed in uh the the more expensive uh this forward pass will be of this neural network but not by much so you should
（或类似数量）。所以对于每个标记，这里发生的计算量是有限的，你可以把这个计算量想象成非常小。并且这个计算量对于序列中的每个标记来说几乎是固定的。实际上并非完全如此，因为输入的标记越多，这个神经网络的前向传递计算成本就越高，但增加的幅度不大。所以你应该……
(1:49:09) think of this uh and I think as a good model to have in mind this is a fixed amount of compute that's going to happen in this box for every single one of these tokens and this amount of compute Cann possibly be too big because there's not that many layers that are sort of going from the top to bottom here there's not that that much computationally that will happen here and so you can't imagine the model to to basically do arbitrary computation in a single forward pass to get a single token and so what that means is that we
这样理解，我觉得可以把它想象成一个很好的模型概念：对于每个标记，在这个“盒子”（指神经网络的计算过程）里发生的计算量是固定的。这个计算量不可能太大，因为从顶部到底部并没有那么多层，这里不会发生那么多计算。所以你不能想象模型在一次前向传递中为了得到一个标记就进行任意复杂的计算。这意味着我们……
(1:49:34) actually have to distribute our reasoning and our computation across many tokens because every single token is only spending a finite amount of computation on it and so we kind of want to distribute the computation across many tokens and we can't have too much computation or expect too much computation out of of the model in any single individual token because there's only so much computation that happens per token okay roughly fixed amount of computation here so that's why this answer here is significantly worse and the reason for
实际上必须将推理和计算分布到多个标记上，因为每个标记只能分配到有限的计算量。所以我们要把计算分布到多个标记上，不能在任何单个标记上期望模型进行过多计算，因为每个标记的计算量是有限的。这里的计算量大致是固定的。这就是为什么这个答案明显更差，原因是……
(1:50:08) that is Imagine going from left to right here um and I copy pasted it right here the answer is three Etc imagine the model having to go from left to right emitting these tokens one at a time it has to say or we're expecting to say "the answer is space dollar sign" and then right here we're expecting it to basically cram all of the computation of this problem into this single token it has to emit the correct answer three and then once

we've emitted the answer three we're expecting it to say all these tokens but at this point we've already
想象一下从左到右的过程。我把答案复制粘贴到这里，“答案是 3 等等”。想象一下模型必须从左到右一次输出一个标记，它必须说，或者我们期望它说 “答案是 空格 美元符号”，然后在这里，我们期望它把这个问题的所有计算都塞进这个单一标记里，它必须输出正确答案 3，然后一旦输出了答案 3，我们又期望它说出后面所有的标记。但此时我们已经……
(1:50:41) prod produced the answer and it's already in the context window for all these tokens that follow so anything here is just um kind of post Hawk justification of why this is the answer um because the answer is already created it's already in the token window so it's it's not actually being calculated here um and so if you are answering the question directly and immediately you are training the model to to try to basically guess the answer in a single token and that is just not going to work because of the finite amount of
生成了答案，而且这个答案已经在后续所有标记的上下文窗口中了。所以这里后面的内容只是对为什么这是答案的一种事后解释，因为答案已经生成了，它已经在标记窗口中了，所以这里实际上并没有在进行计算。所以如果你直接立即回答问题，你就是在训练模型尝试在一个标记中猜出答案，而这是行不通的，因为每个标记的计算量是有限的。
(1:51:11) computation that happens per token that's why this answer on the right is significantly better because we are Distributing this computation across the answer we're actually getting the model to sort of slowly come to the answer from the left to right we're getting intermediate results we're saying "okay the total cost of oranges is four so 13 - 4 is 9" and so we're creating intermediate calculations and each one of these calculations is by itself not that expensive and so we're actually basically kind of guessing a little bit
每个标记的计算量有限。这就是为什么右边的答案明显更好，因为我们在答案中分布了计算过程，实际上是让模型从左到右逐步得出答案。我们得到了中间结果，比如我们说 “好的，橙子的总成本是 4，所以 13 减 4 等于 9”，所以我们创建了中间计算步骤，而且每个这样的计算步骤本身的计算量并不大。所以我们实际上是在大致……
(1:51:40) the difficulty that the model is capable of in any single one of these individual tokens and there can never be too much work in any one of these tokens computationally because then the model won't be able to do that later at test time and so we're teaching the model here to spread out its reasoning and to spread out its computation over the tokens and in this way it only has very simple problems in each token and they can add up and then by the time it's near the end it has all the previous results in its working memory and it's
猜测模型在单个标记中能够处理的难度，并且任何一个标记的计算量都不能太大，因为否则模型在测试时就无法完成计算。所以我们在这里是在教模型将推理和计算分散到各个标记上，通过这种方式，每个标记只处理非常简单的问题，这些问题的结果可以累加。然后当接近答案时，它的工作记忆中已经有了之前所有的结果，这样……
(1:52:12) much easier for it to determine that the answer is and here it is three so this is a significantly better label for our computation this would be really bad and is teaching the model to try to do all the computation in a single token and it's really bad so uh that's kind of like an interesting thing to keep in mind is in your prompts uh usually don't have to think about it explicitly because uh the people at open AI have labelers and so on that actually worry about this and they make sure that the answers are
它就更容易确定答案，在这里答案就是 3。所以对于我们的计算来说，这是一个明显更好的标注方式。而另一种方式会很糟糕，它会让模型尝试在一个标记中完成所有计算，这真的很不好。所以这是一个在设计提示时需要记住的有趣点。通常你不必明确考虑这个问题，因为 OpenAI 的工作人员和标注员等会关注这个问题，他们会确保答案……
(1:52:42) spread out and so actually open AI will kind of like do the right thing so when I ask this question for chat GPT it's actually going to go very slowly it's going to be like "okay let's define our variables set up the equation" and it's kind of creating all these intermediate results these are not for you these are for the model if the model is not creating these intermediate results for itself it's not going to be able to reach three I also wanted to show you that it's possible to be a bit mean to the model uh we can just ask for
是分散展开的。实际上 OpenAI 会处理好这些。所以当我在 ChatGPT 中问这个问题时，它实际上会慢慢地思考，它可能会说 “好的，让我们定义变量，建立方程”，它会生成所有这些中间结果。这些中间结果不是给你的，是为了模型自身计算的。如果模型不自己生成这些中间结果，它就无法得出 3 这个答案。我还想告诉你，我们可以故意为难一下模型。我们可以要求……
(1:53:08) things so as an example I said I gave it the exact same uh prompt and I said "answer the question in a single token just immediately give me the answer nothing else" and it turns out that for this simple um prompt here it actually was able to do it in

single go so it just created a single I think this is two tokens right uh because the dollar sign is its own token so basically this model didn't give me a single token it gave me two tokens but it still produced the correct answer and it did that in a single forward pass of the

一些特殊情况。比如，我给它相同的提示，然后说 “用一个标记回答问题，直接给我答案，不要其他内容”。结果对于这个简单的提示，它实际上一次就做到了。我想它只生成了两个标记，对吧？因为美元符号是单独的一个标记。所以基本上这个模型没有给我一个标记，而是给了两个，但它仍然给出了正确答案，并且是在一次网络前向传递中完成的。

(1:53:38) network now that's because the numbers here I think are very simple and so I made it a bit more difficult to be a bit mean to the model so I said "Emily buys 23 apples and 177 oranges" and then I just made the numbers a bit bigger and I'm just making it harder for the model I'm asking it to more computation in a single token and so I said the same thing and here it gave me five and five is actually not correct so the model failed to do all of this calculation in a single forward pass of the network it failed to go from the input tokens and

这是因为我觉得这里的数字很简单。所以我为了刁难它，把问题变得更难了一些。我说 “艾米丽买了 23 个苹果和 177 个橙子”，把数字变大了，让模型的计算难度增加。我还是要求它用一个标记回答，然后它给出了 5，而 5 实际上是不正确的。所以模型没能在一次网络前向传递中完成所有这些计算，它无法从输入标记开始……

(1:54:08) then in a single forward pass of the network single go through the network it couldn't produce the result and then I said "okay now don't worry about the the token limit and just solve the problem as usual" and then it goes all the intermediate results it simplifies and every one of these intermediate results here and intermediate calculations is much easier for the model and um it sort of it's not too much work per token all of the tokens here are correct and it arrives at the solution which is seven and I just couldn't squeeze all of this work

在一次网络前向传递中得出结果。然后我说 “好吧，现在别管标记限制了，像平常一样解决这个问题”，然后它给出了所有中间结果，进行了化简。这里的每一个中间结果和中间计算步骤对模型来说都容易多了。每个标记的计算量都不算太大，这里所有的标记都是正确的，它得出了正确答案 7。我之前只是不让它把所有计算都压缩在一个标记里。

(1:54:38) it couldn't squeeze that into a single forward passive Network so I think that's kind of just a cute example and something to kind of like think about and I think it's kind of again just elucidative in terms of how these uh models work the last thing that I would say on this topic is that if I was in practi is trying to actually solve this in my day - to - day life I might actually not uh trust that the model that all the intermediate calculations correctly here so actually probably what I do is something like this I would come here

它确实无法在一次前向传递中完成所有计算。我觉得这是个很有趣的例子，值得思考。这也进一步说明了这些模型的工作原理。关于这个话题我最后想说的是，如果在日常生活中我真的要解决这类问题，我可能不会完全相信模型能把所有中间计算都做对。所以实际上我可能会这样做，我会……

(1:55:02) and I would say "use code" and uh that's because code is one of the possible tools that chachy PD can use and instead of it having to do mental arithmetic like this mental arithmetic here I don't fully trust it and especially if the numbers get really big there's no guarantee that the model will do this correctly any one of these intermediates steps might in principle fail we're using neural networks to do mental arithmetic uh kind of like you doing mental arithmetic in your brain it might just like uh screw up some of the

说 “使用代码”。这是因为代码是 ChatGPT 可以使用的工具之一。我不完全相信它能像这样进行心算，尤其是当数字变得很大时，不能保证模型能正确计算。这里的任何一个中间步骤都有可能出错，我们用神经网络来做心算，就像你在大脑里做心算一样，它可能会在某些中间结果上出错。

(1:55:31) intermediate results it's actually kind of amazing that it can even do this kind of mental arithmetic I don't think I could do this in my head but basically the model is kind of like doing it in its head and I don't trust that so I wanted to use tools so you can say stuff like "use code" and uh I'm not sure what happened there "use code" and so um like I mentioned there's a special tool and the uh the model can write code and I can inspect that this code is correct and then uh it's not relying on its mental arithmetic it is

实际上，模型能进行这样的心算已经有点不可思议了，我觉得我自己都没法在脑子里做这样的计算。但基本上模型就像在脑子里进行计算一样，而我不太信任它。所以我想用工具，你可以像这样说 “使用代码”。我不太确定这里发生了什么，就是说 “使用代码”。就像我提到的，有一个特殊工具，模型可以编写代码，我可以检查代码是否正确。这样它就不用依赖心算，而是……

(1:56:04) using the python interpreter which is a very simple programming language to basically uh write out the code that calculates the result and I would personally trust this a lot more because this came out of a Python program which I think has a lot more correctness guarantees than the mental arithmetic of a language model uh so just um another kind of uh potential hint that if you have these kinds of problems uh you may want to basically just uh ask the model to use the code interpreter and just like we saw with the web search the

使用 Python 解释器，这是一种非常简单的编程语言，用它来编写计算结果的代码。我个人会更相信这种方式，因为这是由 Python 程序得出的结果，我认为它比语言模型的心算更有正确性保证。所以这只是另一个潜在的提示，如果你遇到这类问题，你可能基本上只想让模型使用代码解释器。就像我们看到的网络搜索一样……

(1:56:31) model has special uh kind of tokens for calling uh like it will not actually generate these tokens from the language model it will write the program and then it actually sends that program to a different sort of part of the computer that actually just runs that program and brings back the result and then the model gets access to that result and can tell you that "okay the cost of each apple is seven" um so that's another kind of tool and I would use this in practice for yourself and it's um yeah it's just uh less error

模型有特殊的标记来调用（这个功能）。实际上，它不是从语言模型中生成这些标记，而是编写程序，然后把程序发送到计算机的另一个部分，这个部分会运行程序并返回结果。然后模型获取这个结果并告诉你 “好的，每个苹果的价格是 7 美元”。所以这是另一种工具，在实际使用中我会用这种方法，它确实…… 出错的可能性更小。

(1:57:02) prone I would say so that's why I called this section "models need tokens to think distribute your competition across many tokens ask models to create intermediate results or whenever you can lean on tools and Tool use instead of allowing the models to do all of the stuff in their memory so if they try to do it all in their memory I don't fully trust it and prefer to use tools whenever possible" I want to show you one more example of where this actually comes up and that's in counting so models actually are not very good at counting

我想说它更不容易出错。这就是为什么我把这部分内容称为 “模型需要标记来思考，将计算分布到多个标记上，让模型生成中间结果，或者尽可能依靠工具，而不是让模型在内存中完成所有工作。因为如果让它们在内存中完成所有工作，我不太信任，所以尽可能使用工具”。我再给你展示一个实际会遇到这种情况的例子，那就是计数。实际上，模型在计数方面并不擅长。

(1:57:30) for the exact same reason you're asking for way too much in a single individual token so let me show you a simple example of that um "how many dots are below" and then I just put in a bunch of dots and Chach says "there are" and then it just tries to solve the problem in a single token so in a single token it has to count the number of dots in its context window um and it has to do that in the single forward pass of a network and a single forward pass of a network as we talked about there's not that much computation

原因和前面一样，就是在单个标记上要求它做的太多了。我给你举个简单的例子，比如 “下面有多少个点”，然后我放了一堆点。ChatGPT 回答 “有……”，然后它试图用一个标记来解决这个问题。所以在一个标记里，它必须数出上下文窗口中的点数，而且必须在一次网络前向传递中完成。就像我们说过的，一次网络前向传递中能进行的计算量并不多。

(1:58:00) that can happen there just think of that as being like very little competation that happens there so if I just look at what the model sees let's go to the LM go to tokenizer it sees uh this "how many dots are below" and then it turns out that these dots here this group of I think 20 dots is a single token and then this group of whatever it is is another token and then for some reason they break up as this so I don't actually this has to do with the details of the tokenizer but it turns out that these um the model basically sees the

所以，想想看，那里能进行的计算量非常小。如果我看看模型 “看到” 的内容，我们打开语言模型的标记器。它看到 “下面有多少个点”，然后结果是，这里的这些点，我觉得这一组 20 个点是一个标记，然后另一组不管有多少个点又是一个标记。然后不知为什么它们是这样划分的。我不太清楚，这和标记器的细节有关，但结果是，模型基本上看到的是……

(1:58:34) token ID this this this and so on and then from these token IDs it's expected to count the number and spoiler alert is not 161 it's actually I believe 177 so here's what we can do instead uh we can say "use code" and you might expect that like why should this work and it's actually kind of subtle and kind of interesting so when I say "use code" I actually expect this to work let's see okay 177 is correct so what happens here is I've actually it doesn't look like it but I've broken down the problem into a problems that are easier for the model I

这些标记的 ID，像这样一个一个的，然后它要根据这些标记 ID 来数数量。剧透一下，答案不是 161，实际上我觉得应该是 177。所以我们可以这样做，我们说 “使用代码”。你可能会想，为什么这样就行得通呢？这其实有点微妙，也挺有意思的。当我说 “使用代码” 时，我其实期待它能奏效。我们看看，好的，177 是正确的。这里发

生的事情是，虽然看起来不像，但实际上我把这个问题分解成了对模型来说更容易的问题。我……
(1:59:10) know that the model can't count it can't do mental counting but I know that the model is actually pretty good at doing copy pasting so what I'm doing here is when I say "use code" it creates a string in Python for this and the task of basically copy pasting my input here to here is very simple because for the model um it sees this string of uh it sees it as just these four tokens or whatever it is so it's very simple for the model to copy paste those token IDs and um kind of unpack them into Dots here and so it creates this string and
知道模型不会数数，它没办法心算点数，但我知道模型其实很擅长复制粘贴。所以我这里的做法是，当我说“使用代码”时，它会用 Python 创建一个字符串。对模型来说，把我这里的输入复制粘贴到那里的任务非常简单，因为它把这个字符串看成是，比如，就这四个标记之类的。所以对模型来说，复制粘贴这些标记 ID 并把它们解析成点是很简单的。所以它创建了这个字符串，然后……
(1:59:48) then it calls python routine. count and then it comes up with the correct answer so the python interpreter is doing the counting it's not the models mental arithmetic doing the counting so it's again a simple example of um models need tokens to think don't rely on their mental arithmetic and um that's why also the models are not very good at counting if you need them to do counting tasks always ask them to lean on the tool now the models also have many other little cognitive deficits here and there and these are kind of like sharp edges of
调用 Python 的计数函数，然后得出了正确答案。所以是 Python 解释器在做计数工作，而不是模型的心算在计数。这又是一个简单的例子，说明模型需要标记来思考，不要依赖它们的心算。这就是为什么模型在计数方面不太擅长。如果你需要它们完成计数任务，一定要让它们借助工具。现在，模型在其他方面也有很多小的认知缺陷，这些有点像是这项技术的……
(2:00:17) the technology to be kind of aware of over time so as an example the models are not very good with all kinds of spelling related tasks they're not very good at it and I told you that we would loop back around to tokenization and the reason to do for this is that the models they don't see the characters they see tokens and they their entire world is about tokens which are these little text chunks and so they don't see characters like our eyes do and so very simple character level tasks often fail so for example uh I'm giving it a string
一些需要注意的地方。比如，模型在各种拼写相关的任务上表现都不太好。我之前说过我们会再回到标记化这个话题，原因就是模型看不到字符，它们看到的是标记，它们的整个“世界”都是由这些小文本块（标记）构成的。所以它们不像我们的眼睛那样能看到字符，所以非常简单的字符级任务它们经常做不好。例如，我给它一个字符串……
(2:00:48) "ubiquitous" and I'm asking it to print only every third character starting with the first one so we start with "U" and then we should go every third so every so 1 2 3 "Q" should be next and then Etc so this I see is not correct and again my hypothesis is that this is again Dental arithmetic here is failing number one a little bit but number two I think the the more important issue here is that if you go to Tik tokenizer and you look at "ubiquitous" we see that it is three tokens right so you and I see "ubiquitous" and we can easily
“ubiquitous”，让它从第一个字符开始，每隔两个字符打印一个。所以我们从“U”开始，然后每隔两个字符，也就是 1、2、3，下一个应该是“Q”，以此类推。我发现它给出的结果不正确。我猜原因一是这里有点像心算出错了，二是我认为更重要的原因是，如果你用 tiktokenizer 查看“ubiquitous”，会发现它被分成了三个标记，对吧？你和我看到“ubiquitous”时，我们可以很容易地……
(2:01:21) access the individual letters because we kind of see them and when we have it in the working memory of our visual sort of field we can really easily index into every third letter and I can do that task but the models don't have access to the individual letters they see this as these three tokens and uh remember these models are trained from scratch on the internet and all these token uh basically the model has to discover how many of all these different letters are packed into all these different tokens and the reason we even use tokens is
访问每个单独的字母，因为我们能看到这些字母，当这个单词在我们的视觉工作记忆中时，我们可以很容易地每隔两个字母选取一个。我能完成这个任务，但模型无法访问单个字母，它们看到的是这三个标记。要记住，这些模型是在互联网数据上从头开始训练的，对于所有这些标记，基本上模型必须去发现每个标记里包含多少个不同的字母。我们使用标记的原因……
(2:01:49) mostly for efficiency uh but I think a lot of people areed interested to delete tokens entirely like we should really have character level or bite level models it's just that that would create very long sequences and people don't know how to deal with that right now so while we have the token World any kind of spelling tasks are not actually expected to work super well so because I know that spelling is not a strong suit because of tokenization I can again Ask it to lean On Tools so I can just say "use code" and I

would again expect this

主要是为了效率。但我觉得很多人都对完全抛弃标记感兴趣，比如我们真的应该有字符级或字节级的模型，只是那样会生成非常长的序列，而现在人们还不知道如何处理。所以在我们使用标记的情况下，任何拼写任务实际上都不太可能完成得很好。因为我知道由于标记化的原因，拼写不是模型的强项，所以我又可以让它借助工具。我可以直接说 “使用代码”，我也会期待……

(2:02:16) to work because the task of copy pasting "ubiquitous" into the python interpreter is much easier and then we're leaning on python interpreter to manipulate the characters of this string so when I say "use code" "ubiquitous" yes it indexes into every third character and the actual truth is "uqs" uh which looks correct to me so um again an example of spelling related tasks not working very well a very famous example of that recently is "how many R are there in strawberry" and this went viral many times and basically the

这样能行得通。因为把 “ubiquitous” 复制粘贴到 Python 解释器中的任务要容易得多，然后我们依靠 Python 解释器来操作这个字符串的字符。所以当我说 “使用代码”，处理 “ubiquitous” 时，它确实每隔两个字符选取一个，实际结果是 “uqs”，我觉得这个结果是正确的。这又是一个拼写相关任务表现不好的例子。最近一个很有名的例子是 “strawberry 里有几个 r”，这个问题多次在网上引起热议。基本上……

(2:02:52) models now get it correct they say there are three Rs in Strawberry but for a very long time all the state-of-the-art models would insist that there are only two Rs in strawberry and this caused a lot of you know Ruckus because is that a word I think so because um it just kind of like why are the models so brilliant and they can solve math Olympiad questions but they can't like count Rs in strawberry and the answer for that again is I've got built up to it kind of slowly but number one the models don't see characters they see tokens and

现在模型能答对这个问题了，它们说 “strawberry” 里有三个 “r”。但在很长一段时间里，所有最先进的模型都坚持认为 “strawberry” 里只有两个 “r”。这引起了很多争议，我觉得 “ruckus” 这个词用在这里挺合适的。因为人们会想，为什么模型这么厉害，能解数学竞赛题，却连 “strawberry” 里有几个 “r” 都数不清楚呢？答案我之前也慢慢提到过，一是模型看不到字符，它们看到的是标记；二是……

(2:03:21) number two they are not very good at counting and so here we are combining the difficulty of seeing the characters with the difficulty of counting and that's why the models struggled with this even though I think by now honestly I think open I may have hardcoded the answer here or I'm not sure what they did but um uh but this specific query now works so models are not very good at spelling and there there's a bunch of other little sharp edges and I don't want to go into all of them I just want to show you a few examples of things to be aware

二是它们不太擅长计数。所以我们把识别字符的困难和计数的困难结合在了一起，这就是为什么模型在这个问题上会有困难。不过说实话，我觉得现在 OpenAI 可能在这里硬编码了答案，我不确定他们具体做了什么，但这个特定的查询现在能得到正确结果了。所以模型在拼写方面不太擅长，而且还有很多其他小的问题，我不想全部讲完，只是想给你展示一些需要注意的例子。

(2:03:50) of and uh when you're using these models in practice I don't actually want to have a comprehensive analysis here of all the ways that the models are kind of like falling short I just want to make the point that there are some Jagged edges here and there and we've discussed a few of them and a few of them make sense but some of them also will just not make as much sense and they're kind of like you're left scratching your head even if you understand in - depth how these models work and and good example of that recently is the following uh the

当你在实际使用这些模型时。我实际上并不想在这里全面分析模型所有的不足之处，我只是想指出模型存在一些问题，我们已经讨论了其中一些，有些问题很好理解，但有些问题即使你深入了解模型的工作原理，也会觉得难以理解。最近有一个很好的例子，就是下面这个……

(2:04:16) models are not very good at very simple questions like this and uh this is shocking to a lot of people because these math uh these problems can solve complex math problems they can answer PhD grade physics chemistry biology questions much better than I can but sometimes they fall short in like super simple problems like this so here we go "9.11 is bigger than 9.

模型在处理这类非常简单的问题时表现不佳，这让很多人感到震惊。因为这些模型可以解决复杂的数学问题，回答博士水平的物理、化学、生物问题，比我厉害得多，但有时在处理像这样非常简单的问题时却表现很差。比如这个问题：“9.11 比 9.

(2:04:38) 9"and it justifies it in some way but obviously and then at the end"okay it actually it flips its decision later"so um I don't believe that this is very reproducible sometimes it flips around its answer sometimes gets it right sometimes get it get it wrong uh let's try again"okay even though it might look larger" okay so here it doesn't even correct itself in the end if you ask many times sometimes it gets it right too but how is it that the model can do so great at Olympiad grade problems but then fail on very simple problems like

9 大”，它还给出了某种解释，但显然是错误的，然后最后又说 “好吧，实际上它后来改变了判断”。我觉得这个结果不太具有可重复性，有时它的答案会变来变去，有时答对，有时答错。我们再试一次，“好吧，尽管它看起来可能更大”，好的，这里它最后都没有纠正自己的错误。如果你多次提问，有时它也能答对。但为什么模型能在奥林匹克竞赛级别的难题上表现出色，却在像这样非常简单的问题上出错呢？

(2:05:12) this and uh I think this one is as I mentioned a little bit of a head scratcher it turns out that a bunch of people studied this in depth and I haven't actually read the paper uh but what I was told by this team was that when you scrutinize the activations inside the neural network when you look at some of the features and what what features turn on or off and what neurons turn on or off uh a bunch of neurons inside the neural network light up that are usually associated with Bible verses U and so I think the model is kind of

我觉得这个问题，就像我提到的，有点让人摸不着头脑。原来有很多人深入研究过这个问题，我实际上还没读过相关论文。但这个团队告诉我，当你仔细观察神经网络内部的激活情况，查看一些特征，看哪些特征被激活或关闭，哪些神经元被激活或关闭时，会发现神经网络里有很多通常与圣经经文相关的神经元被激活了。所以我觉得模型有点……

(2:05:43) like reminded that these almost look like Bible verse markers and in a Bible verse setting 9.11 would come after 9.9 and so basically the model somehow finds it like cognitively very distracting that in Bible verses 9.11 would be greater um even though here it's actually trying to justify it and come up to the answer with a math it still ends up with the wrong answer here so it basically just doesn't fully make sense and it's not fully understood and um there's a few Jagged issues like that so that's why treat this as a as what it is

像是被这些数字提醒了，它们看起来有点像圣经经文的标记，在圣经经文的语境中，9.11 会排在 9.9 后面。所以基本上，模型在认知上会被这个干扰，认为在圣经经文的语境里 9.11 更大，尽管它实际上试图用数学来解释并得出答案，但最后还是错了。所以这个问题基本上不太能说得通，也没有被完全理解。还有一些类似这样奇怪的问题。这就是为什么要把模型当作它实际的样子来看待……

(2:06:17) which is a St stochastic system that is really magical but that you can't also fully trust and you want to use it as a tool not as something that you kind of like letter rip on a problem and copypaste the results okay so we have now covered two major stages of training of large language models we saw that in the first stage this is called the pre-training stage we are basically training on internet documents and when you train a language model on internet documents you get what's called a base model and it's basically an internet

它是一个随机系统，虽然很神奇，但你不能完全信任它。你应该把它当作一个工具来使用，而不是在遇到问题时完全依赖它，直接照搬它给出的结果。好的，现在我们已经介绍了大语言模型训练的两个主要阶段。我们看到，第一个阶段叫做预训练阶段，基本上就是在互联网文档上进行训练。当你在互联网文档上训练语言模型时，会得到一个所谓的基础模型，它基本上是一个互联网……

(2:06:46) document simulator right now we saw that this is an interesting artifact and uh this takes many months to train on thousands of computers and it's kind of a lossy compression of the internet and it's extremely interesting but it's not directly useful because we don't want to sample internet documents we want to ask questions of an AI and have it respond to our questions so for that we need an assistant and we saw that we can actually construct an assistant in the process of a post training and specifically in the process

文档模拟器。我们看到这是一个很有趣的产物，它需要在数千台计算机上花费数月时间进行训练，它有点像是对互联网的一种有损压缩，非常有趣，但它并不能直接发挥作用，因为我们不想只是采样互联网文档，我们希望向人工智能提问并得到回答。所以为此我们需要一个助手，我们看到实际上可以在后训练过程中构建一个助手，具体来说是在……

(2:07:17) of supervised fine-tuning as we call it so in this stage we saw that it's algorithmically identical to pre-training nothing is going to change the only thing that changes is the data set so instead of Internet documents we now want to create and curate a very nice data set of conversations so we want Millions conversations on all kinds of diverse topics between a human and an assistant and fundamentally these conversations are created by humans so humans write the prompts and humans write the ideal response responses and

我们称之为监督微调的过程中。在这个阶段，我们看到它在算法上和预训练是一样的，没有什么变化，唯一改变的是数据集。所以我们不再使用互联网文档，而是要创建和整理一个非常好的对话数据集。我们需要数百万个关于各种不同主题的人类与助手之间的对话。从根本上说，这些对话是由人类创建的，人类编写问题，人类编写理想的回答，并且……

(2:07:52) they do that based on labeling documentations now in the modern stack it's not actually done fully and manually by humans right they actually now have a lot of help from these tools so we can use language models um to help us create these data sets and that's done extensively but fundamentally it's all still coming from Human curation at the end so we create these conversations that now becomes our data set we fine tune on it or continue training on it and we get an assistant and then we kind of shifted gears and started talking

他们根据标注文档来做这些。在现代的技术流程中，实际上并不完全是由人类手动完成这些工作的，对吧？现在人们实际上从这些工具中得到了很多帮助。我们可以使用语言模型来帮助创建这些数据集，而且应用非常广泛，但从根本上说，最终还是离不开人类的整理。所以我们创建这些对话，它们现在成为了我们的数据集，我们在上面进行微调或继续训练，从而得到一个助手。然后我们转换话题，开始讨论……

(2:08:22) about some of the kind of cognitive implications of what this assistant is like and we saw that for example the assistant will hallucinate if you don't take some sort of mitigations towards it so we saw that hallucinations would be common and then we looked at some of the mitigations of those hallucinations and then we saw that the models are quite impressive and can do a lot of stuff in their head but we saw that they can also Lean On Tools to become better so for example we can lo lean on a web search in order to hallucinate less and to

这个助手的一些认知方面的特点。我们看到，例如，如果不采取一些缓解措施，助手就会产生幻觉。我们看到幻觉是很常见的，然后我们探讨了一些缓解幻觉的方法。然后我们看到模型非常强大，能在“头脑”里完成很多任务，但我们也看到它们还可以借助工具变得更好。比如，我们可以借助网络搜索来减少幻觉，并且……

(2:08:51) maybe bring up some more um recent information or something like that or we can lean on tools like code interpreter so the code can so the llm can write some code and actually run it and see the results so these are some of the topics we looked at so far um now what I'd like to do is I'd like to cover the last and major stage of this Pipeline and that is reinforcement learning so reinforcement learning is still kind of thought to be under the umbrella of posttraining uh but it is the last third major stage and

可能获取一些更新的信息之类的，或者我们可以借助代码解释器这样的工具，这样大语言模型就可以编写代码并实际运行它，查看结果。这些就是我们目前讨论过的一些话题。现在我想介绍这个流程的最后一个主要阶段，那就是强化学习。强化学习仍然被认为属于后训练的范畴，但它是后训练的第三个主要阶段，而且……

(2:09:22) it's a different way of training language models and usually follows as this third step so inside companies like open AI you will start here and these are all separate teams so there's a team doing data for pre-training and a team doing training for pre-training and then there's a team doing all the conversation generation in a in a different team that is kind of doing the supervis fine tuning and there will be a team for the reinforcement learning as well so it's kind of like a handoff of these models you get your base model the

它是训练语言模型的一种不同方式，通常作为第三步。在像 OpenAI 这样的公司里，你会从这里开始，各个环节由不同的团队负责。有一个团队负责预训练的数据准备，一个团队负责预训练，还有一个团队负责生成对话，另一个团队进行监督微调，也会有一个团队负责强化学习。所以这有点像模型在不同团队之间的交接。你得到基础模型后……

(2:09:51) then you find you need to be an assistant and then you go into reinforcement learning which we'll talk about uh now so that's kind of like the major flow and so let's now focus on reinforcement learning the last major stage of training and let me first actually motivate it and why we would want to do reinforcement learning and what it looks like on a high level so I would now like to try to motivate the reinforcement learning stage and what it corresponds to with something that you're probably familiar with and that

将它改进为助手模型，然后进入强化学习阶段，我们现在就来讨论这个阶段。这就是大致的流程。现在让我们专注于强化学习，这是训练的最后一个主要阶段。首先，我来解释一下为什么要进行强化学习，以及从宏观层面看它是什么样的。我想用一个你可能熟悉的事情来解释强化学习阶段，以及它对应的是什么。

(2:10:17) is basically going to school so just like you went to school to become um really good at something we want to take large language models through school and really what we're doing is um we're um we have a few paradigms of ways of uh giving them knowledge or transferring skills so in particular when we're working with textbooks in school you'll see that there are three major kind of uh pieces of information in these textbooks three classes of information the first thing you'll see is you'll see a lot of exposition um and by the way

这个事情就是上学。就像你上学是为了在某个方面变得很擅长一样，我们也想让大语言模型“上学”。实际上我们所做的是，我们有几种传授知识或技能的模式。特别是当我们在学校使用教科书时，你会发现这些教科书里主要有三类信息。首先，你会看到很多讲解内容。顺便说一下……

(2:10:49) this is a totally random book I pulled from the internet I I think it's some kind of an organic chemistry or something I'm not sure uh but the important thing is that you'll see that most of the text most of it is kind of just like the meat of it is exposition it's kind of like background knowledge Etc as you are reading through the words of this Exposition you can think of that roughly as training on that data so um and that's why when you're reading through this stuff this background knowledge and this all this context
这是我从网上随机找的一本书，我觉得可能是有机化学之类的书，我不太确定。但重要的是，你会发现书中大部分内容都是讲解，有点像背景知识等等。当你阅读这些讲解内容时，你可以大致把这看作是在这些数据上进行训练。所以…… 这就是为什么当你阅读这些内容时，这些背景知识和所有这些上下文……
(2:11:16) information it's kind of equivalent to pre-training so it's it's where we build sort of like a knowledge base of this data and get a sense of the topic the next major kind of information that you will see is these uh problems and with their worked Solutions so basically a human expert in this case uh the author of this book has given us not just a problem but has also worked through the solution and the solution is basically like equivalent to having like this ideal response for an assistant so it's basically the expert is showing us how
信息有点相当于预训练。这是我们建立这些数据的知识库，并对主题有个大致了解的过程。你会看到的下一类主要信息是问题以及它们的解答。基本上，在这种情况下，人类专家，也就是这本书的作者，不仅给了我们问题，还给出了解题过程。这个解题过程基本上就相当于助手的理想回答。所以基本上就是专家在向我们展示如何……
(2:11:50) to solve the problem in it's uh kind of like um in its full form so as we are reading the solution we are basically training on the expert data and then later we can try to imitate the expert um and basically um that's that roughly correspond to having the sft model that's what it would be doing so basically we've already done pre-training and we've already covered this um imitation of experts and how they solve these problems and the third stage of reinforcement learning is basically the practice problems so
完整地解决问题。所以当我们阅读解答时，我们基本上是在专家数据上进行训练，之后我们可以尝试模仿专家。基本上…… 这大致相当于使用监督微调（SFT）模型所做的事情。所以基本上我们已经完成了预训练，也讨论了模仿专家以及他们解决问题的方式。强化学习的第三个阶段基本上就是练习题。所以……
(2:12:24) sometimes you'll see this is just a single practice problem here but of course there will be usually many practice problems at the end of each chapter in any textbook and practice problems of course we know are critical for learning because what are they getting you to do they're getting you to practice uh to practice yourself and discover ways of solving these problems yourself and so what you get in a practice problem is you get a problem description but you're not given the solution but you are given the final
有时你会看到这里只有一道练习题，但当然，在任何一本教科书的每一章结尾通常都会有很多练习题。我们都知道练习题对学习至关重要，因为它们让你做什么呢？它们让你练习，自己去练习并找到解决这些问题的方法。所以在做练习题时，你会得到问题描述，但不会得到答案，不过会告诉你最终的……
(2:12:51) answer answer usually in the answer key of the textbook and so you know the final answer that you're trying to get to and you have the problem statement but you don't have the solution you are trying to practice the solution you're trying out many different things and you're seeing what gets you to the final solution the best and so you're discovering how to solve these problems so and in the process of that you're relying on number one the background information which comes from pre-training and number two maybe a
答案，答案通常在教科书的答案部分。所以你知道自己要努力得出的最终答案，也有问题描述，但没有解题过程。你要尝试解答，尝试很多不同的方法，看看哪种方法最能让你得出最终答案。在这个过程中，你一方面依赖预训练获得的背景知识，另一方面可能会……
(2:13:18) little bit of imitation of human experts and you can probably try similar kinds of solutions and so on so we've done this and this and now in this section we're going to try to practice and so we're going to be given prompts we're going to be given Solutions U sorry the final answers but we're not going to be given expert Solutions we have to practice and try stuff out and that's what reinforcement learning is about okay so let's go back to the problem that we worked with previously just so we have a concrete example to talk
借鉴一些人类专家的做法，尝试类似的解题思路等等。我们已经完成了前面这些阶段，现在在这个部分，我们要尝试练习。我们会得到问题提示，会得到最终答案，但不会得到专家给出的解答过程，我们必须自己去尝试解答。这就是强化学习的意义所在。好的，让我们回到之前讨论过的问题，这样我们就有一个具体的例子来探讨……
(2:13:48) through as we explore sort of the topic here so um I'm here in the Teck tokenizer because I'd also like to well I get a text box which is useful but number two I want to remind you again that we're always working with one - dimensional token sequences and so um I actually like prefer this view because this is like the native view of the llm if that makes sense like this is what it actually sees it sees token IDs right okay so

"Emily buys three apples and two oranges each orange is $2 the total cost of all the fruit is $13 what is the cost
这个话题。我现在在 Teck tokenizer 这里，一方面是因为这里有个文本框很有用，另一方面我想再次提醒你，我们处理的始终是一维标记序列。实际上我更喜欢这个视角，因为这就像是大语言模型的 “原生视角”，如果这么说能理解的话，这就是它实际 “看到” 的东西，它看到的是标记 ID，对吧？好的，“艾米丽买了 3 个苹果和 2 个橙子，每个橙子 2 美元，所有水果的总成本是 13 美元，苹果的单价是……
(2:14:20) of each apple" and what I'd like to what I like you to appreciate here is these are like four possible candidate Solutions as an example and they all reach the answer three now what I'd like you to appreciate at this point is that if I am the human data labeler that is creating a conversation to be entered into the training set I don't actually really know which of these conversations to um to add to the data set some of these conversations kind of set up a system equations some of them sort of like just talk through it in
多少？” 我希望你能注意到，这里有四个可能的候选解答示例，它们都得出了答案 3。此刻我希望你明白的是，如果我是创建对话并将其放入训练集的数据标注员，我其实并不知道该把这些对话中的哪一个添加到数据集中。这些对话中，有些是通过建立方程组来解答，有些只是用文字阐述解答过程，还有些……
(2:14:54) English and some of them just kind of like skip right through to the solution um if you look at chbt for example and you give it this question it defines a system of variables and it kind of like does this little thing what we have to appreciate and uh differentiate between though is um the first purpose of a solution is to reach the right answer of course we want to get the final answer three that is the that is the important purpose here but there's kind of like a secondary purpose as well where here we are also just kind
直接给出答案。例如，如果你在 ChatGPT 中提出这个问题，它会定义变量并建立方程组来解答。不过，我们必须认识到并加以区分的是，解答的首要目的当然是得出正确答案，我们希望得到最终答案 3，这是最重要的目的。但这里还有一个次要目的，那就是……
(2:15:24) of trying to make it like nice uh for the human because we're kind of assuming that the person wants to see the solution they want to see the intermediate steps we want to present it nicely Etc so there are two separate things going on here number one is the presentation for the human but number two we're trying to actually get the right answer um so let's for the moment focus on just reaching the final answer if we're only care if we only care about the final answer then which of these is the optimal or the best prompt um sorry
尽量让解答对人类来说更友好。因为我们假设人们想要看到解答过程，想要看到中间步骤，希望解答呈现得更清晰等等。所以这里有两个不同的目标，一是为了让人类更好理解，二是要得到正确答案。那么目前我们先专注于得到最终答案，如果我们只关心最终答案，那么在这些解答中，哪一个是最优或者说最好的呢？抱歉，这里说的 “prompt” 应该是 “solution”（解答），那么……
(2:15:54) the best solution for the llm to reach the right answer um and what I'm trying to get at is we don't know me as a human labeler I would not know which one of these is best so as an example we saw earlier on when we looked at um the token sequences and the mental arithmetic and reasoning we saw that for each token we can only spend basically a finite number of finite amount of compute here that is not very large or you should think about it that way way and so we can't actually make too big of a leap in any one token is is
对于大语言模型来说，哪一个解答能最有效地让它得出正确答案呢？我想说的是，我们并不知道。作为人类标注员，我也不知道哪一个是最好的。例如，我们之前在研究标记序列以及心算和推理时发现，对于每个标记，我们基本上只能分配有限的计算量，而且这个计算量并不大，你应该这样理解。所以我们不能在任何一个标记上进行太大的计算 “跳跃”，也许可以这么理解。
(2:16:27) maybe the way to think about it so as an example in this one what's really nice about it is that it's very few tokens so it's going to take us very short amount of time to get to the answer but right here when we're doing "30 - 4 IDE 3 equals" right in this token here we're actually asking for a lot of computation to happen on that single individual token and so maybe this is a bad example to give to the llm because it's kind of incentivizing it to skip through the calculations very quickly and it's going to actually make up mistakes make
例如，在这个解答中，好处是它使用的标记很少，所以我们能很快得出答案。但就在这里，当我们进行 “30 - 4 IDE 3 equals”（这里疑似 “30 - 4 ÷ 3 equals”）计算时，在这个标记上，我们实际上要求模型在单个标记上进行大量计算。所以这对大语言模型来说可能不是一个好例子，因为这会促使它快速跳过计算过程，实际上可能会导致错误，会……
(2:16:54) mistakes in this mental arithmetic uh so maybe it would work better to like spread out the spread it out more maybe it would be better to set it up as an equation

maybe it would be better to talk through it we fundamentally don't know and we don't know because what is easy for you or I as or as human labelers what's easy for us or hard for us is different than what's easy or hard for the llm it cognition is different um and the token sequences are kind of like different hard for it and so some of the token sequences here that are trivial

在心算过程中出错。所以也许把计算过程展开会更好，也许建立一个方程更好，也许用文字详细阐述更好。但我们根本不知道哪种更好，因为对于你我这样的人类标注员来说容易或困难的事情，对于大语言模型来说并不一样，它的认知方式不同。而且对它来说，不同的标记序列难度也不同。所以这里有些对我们来说很简单的标记序列……

(2:17:30) for me might be um very too much of a leap for the llm so right here this token would be way too hard but conversely many of the tokens that I'm creating here might be just trivial to the llm and we're just wasting tokens like why waste all these tokens when this is all trivial so if the only thing we care care about is the final answer and we're separating out the issue of the presentation to the human um then we don't actually really know how to annotate this example we don't know what solution to get to the llm because we

对大语言模型来说可能跨度太大。比如这里这个标记对它来说可能太难了，但相反，我在这里创建的很多标记对大语言模型来说可能太简单，我们只是在浪费标记，为什么要在这些简单的内容上浪费这么多标记呢？所以，如果我们只关心最终答案，不考虑向人类展示的问题，那么实际上我们并不知道如何为这个例子进行标注，不知道该给大语言模型提供哪种解答，因为我们……

(2:18:01) are not the llm and it's clear here in the case of like the math example but this is actually like a very pervasive issue like for our knowledge is not lm's knowledge like the llm actually has a ton of knowledge of PhD in math and physics chemistry and whatnot so in many ways it actually knows more than I do and I'm I'm potentially not utilizing that knowledge in its problem solving but conversely I might be injecting a bunch of knowledge in my solutions that the LM doesn't know in its parameters and then those are like sudden leaps

不是大语言模型。在这个数学例子中这一点很明显，但实际上这是一个非常普遍的问题。我们的知识和大语言模型的知识不同，大语言模型实际上拥有大量数学、物理、化学等领域的博士级知识。在很多方面它实际上比我知道的更多，而我在为它提供解答时可能没有利用到它的这些知识。但相反，我在解答中可能会加入一些它在参数中没有的知识，这些对它来说就像是突然的 “跳跃”……

(2:18:33) that are very confusing to the model and so our cognitions are different and I don't really know what to put here if all we care about is the reaching the final solution and doing it economically ideally and so long story short we are not in a good position to create these uh token sequences for the LM and they're useful by imitation to initialize the system but we really want the llm to discover the token sequences that work for it we need to find it needs to find for itself what token sequence reliably gets to the answer

会让模型感到困惑。所以我们的认知方式不同，如果我们只关心得出最终答案，并且理想地、高效地做到这一点，我真的不知道该在这里提供什么。长话短说，我们并不擅长为大语言模型创建这些标记序列。虽然通过模仿来初始化系统是有用的，但我们真正希望的是大语言模型自己发现适合它的标记序列。我们需要让它自己找到什么样的标记序列能够可靠地得出答案。

(2:19:10) given the prompt and it needs to discover that in the process of reinforcement learning and of trial and error so let's see how this example would work like in reinforcement learning okay so we're now back in the huging face inference playground and uh that just allows me to very easily call uh different kinds of models so as an example here on the top right I chose the Gemma 2 2 billion parameter model so two billion is very very small so this is a tiny model but it's okay so we're going to give it um the way that

在给定问题提示的情况下，并且它需要在强化学习和试错的过程中发现这些。那么让我们看看在强化学习中这个例子是如何运作的。好的，我们现在回到 Hugging Face 推理游乐场，它能让我很容易地调用不同类型的模型。例如，我在右上角选择了 Gemma 2，一个 20 亿参数的模型。20 亿参数非常少，所以这是一个很小的模型，但没关系。我们将用以下方式来测试它……

(2:19:41) reinforcement learning will basically work is actually quite quite simple um we need to try many different kinds of solutions and we want to see which Solutions work well or not so we're basically going to take the prompt we're going to run the model and the model generates a solution and then we're going to inspect the solution and we know that the correct answer for this one is $3 and so indeed the model gets it correct it says it's $3 so this is correct so that's just one attempt at DIS solution so now we're

强化学习的基本运作方式其实非常简单。我们需要尝试很多不同的解答方式，看看哪些效果好，哪些不好。所以我们基本上会输入问题提示，运行模型，模型会生成一个解答，然后我们检查这个解答。我们知道这个问题的正确答案是 3 美元，这个模型确实答对了，它说答案是 3 美元，这是正确的。这只是一次解答尝试。现在我们……

(2:20:12) going to delete this and we're going to rerun it again let's try a second attempt so the model solves it in a bit slightly different way right every single attempt will be a different generation because these models are stochastic systems remember that at every single token here we have a probability distribution and we're sampling from that distribution so we end up kind kind of going down slightly different paths and so this is a second solution that also ends in the correct answer now we're going to delete that

要删除这个结果并再次运行。让我们进行第二次尝试。模型这次的解答方式稍有不同，对吧？每次尝试生成的结果都会不同，因为这些模型是随机系统。记住，在每个标记处我们都有一个概率分布，我们从这个分布中进行采样，所以最终会走向略有不同的 “路径”。这是第二个解答，也得出了正确答案。现在我们删除这个结果……

(2:20:39) let's go a third time okay so again slightly different solution but also gets it correct now we can actually repeat this uh many times and so in practice you might actually sample thousand of independent Solutions or even like million solutions for just a single prompt um and some of them will be correct and some of them will not be very correct and basically what we want to do is we want to encourage the solutions that lead to correct answers so let's take a look at what that looks like so if we come back over here here's

再进行第三次尝试。好的，又是一个稍有不同的解答，但也答对了。实际上我们可以多次重复这个过程。在实际操作中，对于一个单一的问题提示，你可能会采样数千个甚至数百万个独立的解答。其中一些会是正确的，一些则不太正确。基本上，我们想要鼓励那些能得出正确答案的解答。我们来看看这是怎么回事。如果我们回到这里，这里有……

(2:21:09) kind of like a cartoon diagram of what this is looking like we have a prompt and then we tried many different solutions in parallel and some of the solutions um might go well so they get the right answer which is in green and some of the solutions might go poorly and may not reach the right answer which is red now this problem here unfortunately is not the best example because it's a trivial prompt and as we saw uh even like a two billion parameter model always gets it right so it's not the best example in that sense but let's just exercise some

一个示意图，展示了大致情况。我们有一个问题提示，然后并行尝试了很多不同的解答。有些解答可能效果很好，得到了正确答案（用绿色表示），有些解答可能效果不好，没有得出正确答案（用红色表示）。不幸的是，这个问题并不是一个很好的例子，因为它太简单了，正如我们所见，即使是一个 20 亿参数的模型也总能答对。从这个角度看，它不是一个好例子。但我们还是发挥一下想象……

(2:21:40) imagination here and let's just suppose that the um green ones are good and the red ones are bad okay so we generated 15 Solutions only four of them got the right answer and so now what we want to do is basically we want to encourage the kinds of solutions that lead to right answers so whatever token sequences happened in these red Solutions obviously something went wrong along the way somewhere and uh this was not a good path to take through the solution and whatever token sequences there were in these Green

假设绿色的解答是好的，红色的解答是不好的。好的，我们生成了 15 个解答，只有 4 个是正确的。现在我们想做的是，基本上就是要鼓励那些能得出正确答案的解答方式。在这些红色解答中出现的任何标记序列，显然在某个地方出了问题，这不是一个好的解题 “路径”。而在这些绿色解答中出现的标记序列……

(2:22:13) Solutions well things went uh pretty well in this situation and so we want to do more things like it in prompts like this and the way we encourage this kind of a behavior in the future is we basically train on these sequences um but these training sequencies now are not coming from expert human annotators there's no human who decided that this is the correct solution this solution came from the model itself so the model is practicing here it's tried out a few Solutions four of them seem to have worked and now the model will kind of

在这些绿色解决方案中，情况进展得相当顺利。所以在类似这样的提示下，我们希望更多地采用类似的做法。我们鼓励这种行为的方式是，基于这些序列进行训练。但现在这些训练序列并非来自专业的人类注释者，没有人为这些解决方案判定正确与否，这些解决方案是模型自己生成的。所以模型在这里进行练习，它尝试了一些解决方案，其中有四个似乎奏效了，现在模型会……

(2:22:44) like train on them and this corresponds to a student basically looking at their Solutions and being like "okay well this one worked really well so this is this is how I should be solving these kinds of problems" and uh here in this example there are many different ways to actually like really tweak the methodology a little bit here but just to give the core idea across maybe it's simplest to just think about take the taking the single best solution out of these four uh like say this one that's why it was yellow uh so this is the the

基于它们进行训练。这就好比一个学生看着自己的解题方法，心想 “好吧，这个方法效果很好，所以我以后就应该用这种方法来解决这类问题”。在这个例子中，实际上有很多不同的方法可以对这种方法进行微调，但为了传达核心思想，也许最简单的就是从这四个中选出最好的一个，比如说这个黄色标记的（假设），这就是……

(2:23:12) solution that not only led to the right answer but may maybe had some other nice properties maybe it was the shortest one or it looked nicest in some ways or uh there's other criteria you could think of as an example but we're going to decide that this the top solution we're going to train on it and then uh the model will be slightly more likely once you do the parameter update to take this path in this kind of a setting in the future but you have to remember that we're going to run many different diverse prompts across lots of math

这个不仅得出了正确答案，可能还具有一些其他优点的解决方案。也许它是最短的，或者在某些方面看起来最好，你还可以想到其他标准。但我们决定把这个作为最佳解决方案，在它的基础上进行训练。然后，一旦你更新参数，模型在未来遇到类似情况时，选择这条“路径”的可能性就会稍微提高。但你要记住，我们会针对大量数学、物理问题以及其他各种可能的问题给出很多不同的提示……

(2:23:42) problems and physics problems and whatever wherever there might be so tens of thousands of prompts maybe have in mind there's thousands of solutions prompt and so this is all happening kind of like at the same time and as we're iterating this process the model is discovering for itself what kinds of token sequences lead it to correct answers it's not coming from a human annotator the the model is kind of like playing in this playground and it knows what it's trying to get to and it's discovering sequences that work for it

可能有成千上万个提示，每个提示又有数千个解决方案。所以这一切几乎是同时发生的。在我们不断重复这个过程时，模型会自己发现什么样的标记序列能让它得出正确答案。这些不是由人类注释者提供的，模型就像是在这个“游乐场”里“玩耍”，它知道自己想要达到的目标，并在发现适合自己的序列。

(2:24:15) uh these are sequences that don't make any mental leaps uh they they seem to work reliably and statistically and uh fully utilize the knowledge of the model as it has it and so uh this is the process of reinforcement learning it's basically a guess and check we're going to guess many different types of solutions we're going to check them and we're going to do more of what worked in the future and that is uh reinforcement learning so in the context of what came before we see now that the sft model the supervised fine

这些序列不会在逻辑上出现跳跃，从统计数据来看，它们似乎能可靠地发挥作用，并且充分利用了模型已有的知识。这就是强化学习的过程，基本上就是不断试错。我们会尝试很多不同类型的解决方案，检查它们的效果，然后在未来更多地采用那些有效的方案，这就是强化学习。结合前面讲的内容，我们现在可以看到，监督微调（SFT）模型……

(2:24:45) tuning model it's still helpful because it still kind of like initializes the model a little bit into to the vicinity of the correct Solutions so it's kind of like a initialization of um of the model in the sense that it kind of gets the model to you know take Solutions like write out Solutions and maybe it has an understanding of setting up a system of equations or maybe it kind of like talks through a solution so it gets you into the vicinity of correct Solutions but reinforcement learning is where everything gets dialed in we really

仍然有帮助，因为它在一定程度上能让模型初步接近正确的解决方案。从某种意义上说，它有点像是对模型的初始化，它能让模型尝试给出解决方案，比如写出解题步骤，也许它对建立方程组有一定的理解，或者能阐述解题思路，从而让模型接近正确答案。但强化学习才是让一切更加精准的阶段，我们真正……

(2:25:14) discover the solutions that work for the model get the right answers we encourage them and then the model just kind of like gets better over time time okay so that is the high Lev process for how we train large language models in short we train them kind of very similar to how we train children and basically the only difference is that children go through chapters of books and they do all these different types of training exercises um kind of within the chapter of each book but instead when we train AIS it's

发现适合模型的解决方案，得到正确答案，并对其加以强化，然后随着时间的推移，模型会不断改进。好的，这就是训练大语言模型的大致过程。简而言之，我们训练大语言模型的方式与培养孩子有相似之处，基本上唯一的区别在于，孩子们通过阅读书本章节，并在每章中完成各种不同类型的练习题来学习，而我们训练人工智能时……

(2:25:41) almost like we kind of do it stage by stage depending on the type of that stage so first what we do is we do pre-training which as we saw is equivalent to uh basically reading all the expository material so we look at all the textbooks at the same time and we read all the exposition and we try to build a knowledge base the second thing then is we go into the sft stage which is really looking at all the fixed uh sort of like solutions from Human Experts of all the different kinds of worked Solutions across all the

几乎是根据不同阶段的特点，分阶段进行训练。首先，我们进行预训练，正如我们所看到的，这相当于阅读所有的讲解性材料。我们同时研读所有的教科书，阅读其中的讲解内容，试图建立一个知识库。然后第二步，我们进入监督微调阶段，这个阶段主要是研究人类专家针对各种不同问题给出的固定解决方案，这些方案涵盖了……

(2:26:12) textbooks and we just kind of get an sft model which is able to imitate the experts but does so kind of blindly it just kind of like does its best guess uh kind of just like trying to mimic statistically the expert behavior and so that's what you get when you look at all the work Solutions and then finally in the last stage we do all the practice problems in the RL stage across all the textbooks we only do the practice

problems and that's how we get the RL model so on a high level the way we train llms is very much equivalent uh to
所有教科书里的各种问题。我们由此得到一个监督微调模型，它能够模仿专家的做法，但这种模仿有点盲目，只是尽力猜测，有点像是从统计意义上模仿专家的行为。这就是我们研究所有解题方案后得到的结果。最后，在最后一个阶段，我们在强化学习阶段针对所有教科书里的问题做练习题，只做练习题，这样我们就得到了经过强化学习的模型。从宏观层面来看，我们训练大语言模型的方式与……
(2:26:43) the process that we train uh that we use for training of children the next point I would like to make is that actually these first two stat ages pre-training and surprise fine-tuning they've been around for years and they are very standard and everyone does them all the different llm providers it is this last stage the RL training that is a lot more early in its process of development and is not standard yet in the field and so um this stage is a lot more kind of early and nent and the reason for that is because I actually skipped over a ton
训练孩子的过程非常相似。我接下来想说的是，实际上前两个阶段，即预训练和监督微调，已经存在多年，非常标准化，所有不同的大语言模型提供商都会采用。而最后这个强化学习阶段在发展过程中还处于更早期的阶段，在该领域尚未形成标准。所以这个阶段还比较新，还在不断发展。原因是我之前实际上略过了很多……
(2:27:13) of little details here in this process the high level idea is very simple it's trial and there learning but there's a ton of details and little math mathematical kind of like nuances to exactly how you pick the solutions that are the best and how much you train on them and what is the prompt distribution and how to set up the training run such that this actually works so there's a lot of little details and knobs to the core idea that is very very simple and so getting the details right here uh is not trivial and so a lot of companies
这个过程中的小细节。从宏观层面看，强化学习的概念很简单，就是试错学习，但在具体实施过程中有很多细节和数学上的细微差别，比如如何挑选最佳解决方案，在这些方案上训练的程度，问题提示的分布情况，以及如何设置训练过程才能让它真正发挥作用。对于这个看似简单的核心概念，有很多细节和需要调整的地方，所以要把这些细节做好并非易事。因此，很多公司……
(2:27:40) like for example open and other LM providers have experimented internally with reinforcement learning fine tuning for llms for a while but they've not talked about it publicly um it's all kind of done inside the company and so that's why the paper from Deep seek that came out very very recently was such a big deal because this is a paper from this company called DC Kai in China and this paper really talked very publicly about reinforcement learning fine training for large language models and how incredibly important it is for large language
比如 OpenAI 和其他大语言模型提供商已经在内部对大语言模型的强化学习微调进行了一段时间的实验，但他们没有公开讨论过。这些工作都是在公司内部进行的。这就是为什么最近 DeepSeek 发表的论文引起了很大的轰动，这篇论文来自中国的 DC Kai 公司，它非常公开地讨论了大语言模型的强化学习微调，以及这对大语言模型来说是多么重要。
(2:28:12) models and how it brings out a lot of reasoning capabilities in the models we'll go into this in a second so this paper reinvigorated the public interest of using RL for llms and gave a lot of the um sort of n-r details that are needed to reproduce their results and actually get the stage to work for large langage models so let me take you briefly through this uh deep seek R1 paper and what happens when you actually correctly apply RL to language models and what that looks like and what that gives you so the first thing I'll scroll
它如何激发出模型的很多推理能力，我们马上会深入探讨。这篇论文重新激发了公众对在大语言模型中使用强化学习的兴趣，并给出了很多详细信息，这些信息对于复现他们的结果、让强化学习阶段在大语言模型中发挥作用非常关键。下面我简要介绍一下 DeepSeek R1 这篇论文，以及当你正确地将强化学习应用于语言模型时会发生什么，会呈现出什么效果，能得到什么。我首先滚动到……
(2:28:41) to is this uh kind of figure two here where we are looking at the Improvement in how the models are solving mathematical problems so this is the accuracy of solving mathematical problems on the a accuracy and then we can go to the web page and we can see the kinds of problems that are actually in these um these the kinds of math problems that are being measured here so these are simple math problems you can um pause the video if you like but these are the kinds of problems that basically the models are being asked to solve and
这里的图二，我们可以看到模型在解决数学问题方面的改进情况。这是解决数学问题的准确率，我们可以访问相关网页，查看这里实际测试的数学问题类型。这些都是简单的数学问题，如果你愿意，可以暂停视频看看。这些就是模型被要求解决的问题类型，并且……
(2:29:08) you can see that in the beginning they're not doing very well but then as you update the model with this many thousands of steps their accuracy kind of continues to

climb so the models are improving and they're solving these problems with a higher accuracy as you do this trial and error on a large data set of these kinds of problems and the models are discovering how to solve math problems but even more incredible than the quantitative kind of results of solving these problems with a higher accuracy is the qualitative means
你可以看到，一开始模型的表现并不好，但随着你对模型进行数千次更新，它们的准确率持续上升。所以，随着在大量这类问题的数据集上进行试错训练，模型在不断改进，解决这些问题的准确率也越来越高。模型在这个过程中逐渐掌握了解决数学问题的方法。但比解决这些问题的准确率提高这一量化结果更令人惊讶的是其定性的表现方式。
(2:29:36) by which the model achieves these results so when we scroll down uh one of the figures here that is kind of interesting is that later on in the optimization the model seems to be uh using average length per response uh goes up up so the model seems to be using more tokens to get its higher accuracy results so it's learning to create very very long Solutions why are these Solutions very long we can look at them qualitatively here so basically what they discover is that the model solution get very very long partially
通过这种方式，模型实现了这些结果。当我们继续往下看，这里有一个有趣的现象，在优化后期，模型的平均每次回复长度似乎在增加。这表明模型似乎在使用更多的标记来获得更高的准确率。它在学习生成很长的解决方案。为什么这些解决方案会很长呢？我们可以从定性的角度来分析。基本上，他们发现模型的解决方案变得很长，部分原因是……
(2:30:07) because so here's a question and here's kind of the answer from the model what the model learns to do um and this is an immerging property of new optimization it just discovers that this is good for problem solving is it starts to do stuff like this "wait wait wait that's Nota moment I can flag here let's reevaluate this step by step to identify the correct sum can be" so what is the model doing here right the model is basically re-evaluating steps it has learned that it works better for accuracy to try out lots of ideas try something from
比如这里有一个问题以及模型给出的答案。模型学会做的事情，这是新的优化过程中出现的一种特性。它发现这样做对解决问题有帮助，它开始这样做："等等，等一下，这里有个问题，我要标记一下。让我们一步一步重新评估，确定正确的总和。" 那么模型在这里到底在做什么呢？它基本上是在重新评估步骤，它认识到尝试很多不同的思路，从不同角度尝试……
(2:30:37) different perspectives retrace reframe backtrack is doing a lot of the things that you and I are doing in the process of problem solving for mathematical questions but it's rediscovering what happens in your head not what you put down on the solution and there is no human who can hardcode this stuff in the ideal assistant response this is only something that can be discovered in the process of reinforcement learning because you wouldn't know what to put here this just turns out to work for the model and it improves its accuracy in
有助于提高准确率，回溯、重新构建思路、倒退检查，它做的很多事情和你我在解决数学问题时的思路是一样的。但它是在重新发现你大脑中的思考过程，而不是简单地照搬解题步骤。没有人能在理想的助手回复中硬性设定这些内容，这只能在强化学习过程中被发现，因为你根本不知道该在这里设定什么，结果发现这样做对模型有效，并且提高了它在……
(2:31:04) problem solving so the model learns what we call these chains of thought in your head and it's an emergent property of the optim of the optimization and that's what's bloating up the response length but that's also what's increasing the accuracy of the problem problem solving so what's incredible here is basically the model is discovering ways to think it's learning what I like to call cognitive strategies of how you manipulate a problem and how you approach it from different perspectives how you pull in some analogies or do
解决问题时的准确率。所以模型学会了我们所说的大脑中的思维链，这是优化过程中出现的一种特性。这就是导致回复长度增加的原因，但同时也提高了问题解决的准确率。这里令人惊讶的是，基本上模型在发现思考的方式，它在学习我所说的认知策略，比如如何处理问题、从不同角度思考问题、如何运用类比或者……
(2:31:34) different kinds of things like that and how you kind of uh try out many different things over time uh check a result from different perspectives and how you kind of uh solve problems but here it's kind of discovered by the RL so extremely incredible to see this emerge in the optimization without having to hardcode it anywhere the only thing we've given it are the correct answers and this comes out from trying to just solve them correctly which is incredible um now let's go back to actually the problem that we've been working with and
进行不同的尝试，随着时间推移从不同角度检查结果，以及如何解决问题。但在这里，这些都是通过强化学习发现的。在优化过程中看到这些出现，而无需在任何地方进行硬编码，这真是太不可思议了。我们只给了它正确答案，而这些都是在努力正确解题的过程中自然出现的，这真的很神奇。现在，让我们回到之前一直在讨论的问题……

(2:32:02) let's take a look at what it would look like uh for uh for this kind of a model what we call reasoning or thinking model to solve that problem okay so recall that this is the problem we've been working with and when I pasted it into chat GPT 40 I'm getting this kind of a response let's take a look at what happens when you give this same query to what's called a reasoning or a thinking model this is a model that was trained with reinforcement learning so this model described in this paper DC car1 is available on chat.dec.com uh so this is

看看对于我们所说的推理或思考模型来说，解决这个问题会是什么样的。回想一下，这就是我们一直在讨论的问题。当我把这个问题粘贴到 ChatGPT 40 中时，得到了这样的回答。现在让我们看看，当把同样的查询输入到所谓的推理或思考模型中会发生什么。这个模型是通过强化学习训练出来的。这篇论文中描述的 DC car1 模型可以在chat.dec.com上使用。所以这是……

(2:32:34) kind of like the company uh that developed is hosting it you have to make sure that the Deep think button is turned on to get the R1 model as it's called we can paste it here and run it and so let's take a look at what happens now and what is the output of the model okay so here's it says so this is previously what we get using basically what's an sft approach a supervised funing approach this is what we get from the RL model okay let me try to figure this out so Emily buys

开发并托管这个模型的公司。你必须确保打开 “Deep think” 按钮才能使用 R1 模型。我们可以把问题粘贴到这里并运行，看看现在会发生什么，模型的输出是什么。好的，它是这样说的。这是我们之前使用监督微调（SFT）方法得到的结果，现在看看从强化学习模型得到的结果。好的，让我试着算一下。艾米丽买了……

(2:33:03) three apples and two oranges each orange cost $2 total is 13 I need to find out blah blah blah so here you you um as you're reading this you can't escape thinking that this model is thinking um is definitely pursuing the solution solution it deres that it must cost $3 and then it says "wait a second let me check my math again to be sure" and then it tries it from a slightly different perspective and then it says "yep all that checks out I think that's the answer I don't see any mistakes let me see if there's another way to

三个苹果和两个橙子，每个橙子 2 美元，总共花费 13 美元，我需要算出…… 等等。当你读这段内容时，你会不禁觉得这个模型在思考。它肯定在努力寻找解决方案，它推断出苹果的价格肯定是 3 美元，然后它说 “等一下，让我再检查一下我的计算，确保没错”，然后它从稍微不同的角度又算了一遍，接着说 “是的，都对，我觉得这就是答案，我没发现任何错误。我再看看有没有其他方法来……

(2:33:34) approach the problem maybe setting up an equation let's let the cost of one apple be $8 then blah blah blah yep same answer so definitely each apple is $3 all right confident that that's correct" and then what it does once it sort of um did the thinking process is it writes up the nice solution for the human and so this is now considering so this is more about the correctness aspect and this is more about the presentation aspect where it kind of like writes it out nicely and uh boxes in the correct answer at the

解决这个问题，也许可以设一个方程。我们设一个苹果的价格为 8 美元，然后…… 是的，答案一样。所以肯定每个苹果是 3 美元。好的，我确定这个答案是正确的”。然后，在完成思考过程后，它为人类整理出了一个不错的解答。所以这里既考虑了…… 这更多是关于答案的正确性方面，而这更多是关于呈现方式方面，它把解答写得很清楚，并且把正确答案框在……

(2:34:05) bottom and so what's incredible about this is we get this like thinking process of the model and this is what's coming from the reinforcement learning process this is what's bloating up the length of the token sequences they're doing thinking and they're trying different ways this is what's giving you higher accuracy in problem solving and this is where we are seeing these "aha" moments and these different strategies and these um ideas for how you can make sure that you're getting the correct answer the last point I wanted to make

底部。这里令人惊讶的是，我们看到了模型的思考过程，这是强化学习过程带来的。这就是导致标记序列变长的原因，它们在思考，尝试不同的方法。这就是在解决问题时能提高准确率的原因，也是我们看到这些 “顿悟” 时刻、不同策略以及确保得到正确答案的思路的地方。我想说的最后一点是……

(2:34:34) is some people are a little bit nervous about putting you know very sensitive data into chat.com because this is a Chinese company so people don't um people are a little bit careful and Cy with that a little bit um deep seek R1 is a model that was released by this company so this is an open source model or open weights model it is available for anyone to download and use you will not be able to like run it in its full um sort of the full model in full Precision you won't run that on a MacBook but uh or like a local device

有些人对在chat.com上输入非常敏感的数据有点担心，因为这是一家中国公司。所以人们会…… 人们对此有点谨慎和担忧。DeepSeek R1 是这家公司发布的模型，这是一个开源模型或者说是开放权重模型，任何人都可以下载使用。你无法在 MacBook 或类似的本地设备上以完整精度运行完整的模型，但是……

(2:35:07) because this is a fairly large model but many companies are hosting the full largest model one of those companies that I like to use is called together. so when you go to together. you sign up and you go to playgrounds you can can select here in the chat deep seek R1 and there's many different kinds of other models that you can select here these are all state-of-the-art models so this is kind of similar to the hugging face inference playground that we've been playing with so far but together.
因为这是一个相当大的模型。不过有很多公司在托管完整的大型模型，我喜欢用的一家公司叫 together. 。当你访问 together. ，注册后进入游乐场板块，你可以在聊天界面中选择 DeepSeek R1，这里还有许多其他不同的模型可供选择，这些都是最先进的模型。这有点类似于我们之前一直在使用的 Hugging Face 推理游乐场，但是 together.……
(2:35:32) a will usually host all the state-of-the-art models so select DT car1 um you can try to ignore a lot of these I think the default settings will often be okay and we can put in this and because the model was released by Deep seek what you're getting here should be basically equivalent to what you're getting here now because of the randomness in the sampling we're going to get something slightly different uh but in principle this should be uh identical in terms of the power of the model and you should be able to see the
通常会托管所有最先进的模型。选择 DT car1，你可以忽略很多设置，我觉得默认设置通常就可以。我们输入问题，因为这个模型是由 DeepSeek 发布的，你在这里得到的结果基本上应该和在chat.dec.com上得到的结果相同。由于采样的随机性，我们会得到略有不同的结果，但原则上，就模型的能力而言，它们应该是一样的，你应该能够看到……
(2:35:59) same things quantitatively and qualitatively uh but uh this model is coming from kind of a an American company so that's deep seek and that's the what's called a reasoning model now when I go back to chat uh let me go to chat here okay so the models that you're going to see in the drop down here some of them like 01 03 mini O3 mini High Etc they are talking about uses Advanced reasoning now what this is referring to uses Advanced reasoning is it's referring to the fact that it was trained by reinforcement learning with
相同的定量和定性结果。但是这个模型来自一家美国公司（此处描述与前文提及的 DeepSeek 为中国公司不符，可能存在表述错误 ）。这就是 DeepSeek，这就是所谓的推理模型。现在我回到聊天界面，我们来看看这里。你在下拉菜单中看到的一些模型，比如 01、03 mini、O3 mini High 等等，它们都提到使用了高级推理。这里所说的使用高级推理，指的是它们是通过强化学习训练的，并且……
(2:36:30) techniques very similar to those of deep C car1 per public statements of opening ey employees uh so these are thinking models trained with RL and these models like GPT 4 or GPT 4 40 mini that you're getting in the free tier you should think of them as mostly sft models supervised fine tuning models they don't actually do this like thinking as as you see in the RL models and even though there's a little bit of reinforcement learning involved with these models and I'll go that into that in a second these are mostly sft models I think you should
根据 OpenAI 员工的公开声明，它们使用的技术与 DeepSeek R1 非常相似。所以这些是通过强化学习训练的思考模型。而像你在免费层级中使用的 GPT 4 或 GPT 4 40 mini，你应该把它们主要看作是监督微调（SFT）模型。它们实际上并不像强化学习模型那样进行思考，尽管这些模型也涉及到一点强化学习，我稍后会讲到这一点。我认为你应该……
(2:37:00) think about it that way so in the same way as what we saw here we can pick one of the thinking models like say 03 mini high and these models by the way might not be available to you unless you pay a Chachi PT subscription of either $20 per month or $200 per month for some of the top models so we can pick a thinking model and run now what's going to happen here is it's going to say "reasoning" and it's going to start to do stuff like this and um what we're seeing here is not exactly the stuff we're seeing here
这样看待它们。就像我们在这里看到的一样，我们可以选择一个思考模型，比如 03 mini high。顺便说一下，除非你每月支付 20 美元或 200 美元（对于一些顶级模型）的 ChatGPT 订阅费用，否则这些模型可能无法使用。我们选择一个思考模型并运行，现在会发生的是，它会显示 “推理中”，然后开始这样做。我们在这里看到的内容和在……
(2:37:29) so even though under the hood the model produces these kinds of uh kind of chains of thought opening ey chooses to not show the exact chains of thought in the web interface it shows little summaries of that of those chains of thought and open kind of does this I think partly because uh they are worried about what's called the distillation risk that is that someone could come in and actually try to imitate those reasoning traces and recover a lot of the reasoning performance by just imitating the reasoning uh chains of
看到的不完全一样。尽管在模型内部会产生这样的思维链，但 OpenAI 选择在网页界面上不展示确切的思维链，而是展示这些思维链的简要总结。我认为 OpenAI 这样做部分原因是…… 他们担心所谓的蒸馏风险，即有人可能会尝

试模仿这些推理过程，仅仅通过模仿推理链就恢复很多推理能力。
(2:37:58) thought and so they kind of hide them and they only show little summaries of them so you're not getting exactly what you would get in deep seek as with respect to the reasoning itself and then they write up the solution so these are kind of like equivalent even though we're not seeing the full under the hood details now in terms of the performance uh these models and deep seek models are currently rly on par I would say it's kind of hard to tell because of the evaluations but if you're paying $200 per month to open AI
所以他们把这些思维链隐藏起来，只展示简要总结。所以就推理本身而言，你在这里得到的和在 DeepSeek 中得到的并不完全一样。然后他们给出解答。所以这些在某种程度上是等效的，尽管我们没有看到完整的底层细节。就性能而言，我得说这些模型和 DeepSeek 的模型目前大致相当，由于评估的原因很难判断。但是如果你每月向 OpenAI 支付 200 美元……
(2:38:25) some of these models I believe are currently they basically still look better uh but deep seek R1 for now is still a very solid choice for a thinking model that would be available to you um sort of um either on this website or any other website because the model is open weights you can just download it so that's thinking models so what is the summary so far well we've talked about reinforcement learning and the fact that thinking emerges in the process of the optimization on when we basically run RL on many math uh and kind of code
我认为目前一些模型看起来仍然更好一些。但 DeepSeek R1 目前仍然是一个非常可靠的思考模型选择，你可以在这个网站或其他网站上使用它，因为这个模型是开放权重的，你可以直接下载。这就是思考模型。到目前为止我们总结一下：我们讨论了强化学习，以及在对许多数学和代码相关问题进行强化学习优化的过程中，思考能力会涌现出来这一事实。
(2:38:57) problems that have verifiable Solutions so there's like an answer three Etc now these thinking models you can access in for example deep seek or any inference provider like together. a and choosing deep seek over there these thinking models are also available uh in chpt under any of the 01 or O3 models but these GPT 4 R models Etc they're not thinking models you should think of them as mostly sft models now if you are um if you have a prompt that requires Advanced reasoning and so on you should probably use some of the
这些问题都有可验证的答案，比如答案是 3 等等。现在这些思考模型，你可以在例如 DeepSeek 或像 together. 这样的推理平台上使用。在这些平台上选择 DeepSeek，这些思考模型在 ChatGPT 中也能找到，比如在 01 或 O3 系列模型中。但像 GPT 4 R 这样的模型，它们不是思考模型，你应该把它们主要看作是监督微调模型。现在，如果你有一个需要高级推理的提示，那么你可能应该使用一些……
(2:39:31) thinking models or at least try them out but empirically for a lot of my use when you're asking a simpler question there's like a knowledge based question or something like that this might be Overkill like there's no need to think 30 seconds about some factual question so for that I will uh sometimes default to just GPT 40 so empirically about 80 90% of my use is just gp4 and when I come across a very difficult problem like in math and code Etc I will reach for the thinking models but then I have to wait a bit longer because
思考模型，或者至少尝试一下。但根据我的经验，在很多情况下，当你问一个比较简单的问题，比如基于知识的问题时，使用思考模型可能有点过头了，没必要为了一个事实性问题思考 30 秒。所以在这种情况下，我有时会默认使用 GPT 40。根据经验，我大约 80 - 90% 的使用场景都只用 GPT 4。当我遇到非常难的问题，比如数学或代码相关的问题时，我会使用思考模型，但那样我就得等更长时间，因为……
(2:40:07) they're thinking um so you can access these on chat on deep seek also I wanted to point out that um AI studio.go.com even though it looks really busy really ugly because Google's just unable to do this kind of stuff well it's like what is happening but if you choose model and you choose here Gemini 2.0 flash thinking experimental 01 21 if you choose that one that's also a a kind of early experiment experimental of a thinking model by Google so we can go here and we can give it the same problem and click run and this is also a
它们在思考。你可以在聊天界面、DeepSeek 上使用这些模型。我还想指出，AI studio.go.com这个平台，尽管它看起来很繁杂、很难看，因为谷歌在这方面做得不太好，让人感觉很奇怪。但是如果你在这个平台上选择模型，选择 Gemini 2.0 flash thinking experimental 01 21，这也是谷歌的一种早期实验性思考模型。我们可以在这里输入同样的问题，点击运行，这也是一个……
(2:40:31) thinking problem a thinking model that will also do something similar and comes out with the right answer here so basically Gemini also offers a thinking model anthropic currently does not offer a thinking model but basically this is kind of like the frontier development of these llms I think RL is kind of like this new exciting stage but getting the details right is difficult and that's why all these models and thinking models are

currently experimental as of 2025 very early 2025 um but this is kind of like
思考相关的问题，这个思考模型也会做出类似的思考并给出正确答案。基本上，Gemini 也提供了一种思考模型，而 Anthropic 目前还没有提供思考模型。这基本上算是大语言模型的前沿发展方向。我认为强化学习是一个令人兴奋的新阶段，但要把细节做好并不容易。这就是为什么截至 2025 年初，所有这些模型，包括思考模型，目前都还处于实验阶段。不过这也算是……
(2:41:01) the frontier development of pushing the performance on these very difficult problems using reasoning that is emerging in these optimizations one more connection that I wanted to bring up is that the discovery that reinforcement learning is extremely powerful way of learning is not new to the field of AI and one place what we've already seen this demonstrated is in the game of Go and famously Deep Mind developed the system alphago and you can watch a movie about it um where the system is learning to play the game of go against top human
利用在这些优化过程中涌现的推理能力来提升在难题上的表现的前沿发展。我还想提到一个关联点，即强化学习是一种极其强大的学习方式，这在人工智能领域并不是新发现。我们已经在一个领域看到了它的显著成效，那就是围棋。著名的 DeepMind 公司开发了 AlphaGo 系统，你可以看一部关于它的电影。在这个系统中，它学习与顶级人类棋手对弈围棋……
(2:41:32) players and um when we go to the paper underlying alphago so in this paper when we scroll down we actually find a really interesting plot um that I think uh is kind of familiar uh to us and we're kind of like we discovering in the more open domain of arbitrary problem solving instead of on the closed specific domain of the game of Go but basically what they saw and we're going to see this in llms as well as this becomes more mature is this is the ELO rating of playing game of Go and this is leas dull an extremely
棋手。当我们查看 AlphaGo 背后的论文时，向下滚动页面，我们会发现一个非常有趣的图表。我觉得这个图表对我们来说有点眼熟，我们现在在更开放的任意问题解决领域进行探索，而不是局限在围棋这个特定封闭领域，但基本上，他们所观察到的现象，随着大语言模型的发展成熟我们也会在其中看到，这个图表展示的是围棋的 ELO 评分，这是李世石，一位极其……
(2:42:07) strong human player and here what they are comparing is the strength of a model learned trained by supervised learning and a model trained by reinforcement learning so the supervised learning model is imitating human expert players so if you just get a huge amount of games played by expert players in the game of Go and you try to imitate them you are going to get better but then you top out and you never quite get better than some of the top top top players of in the game of Go like LEL so you're never going to reach there because
强大的人类棋手。这里他们比较的是通过监督学习训练的模型和通过强化学习训练的模型的实力。监督学习模型是在模仿人类专家棋手，所以如果你获取大量专家棋手的棋局并尝试模仿他们，你的水平会提高，但之后就会达到瓶颈，你永远无法超越像李世石这样的顶级棋手。因为……
(2:42:38) you're just imitating human players you can't fundamentally go beyond a human player if you're just imitating human players but in a process of reinforcement learning is significantly more powerful in reinforcement learning for a game of Go it means that the system is playing moves that empirically and statistically lead to win to winning the game and so alphago is a system where it kind of plays against it itself and it's using reinforcement learning to create rollouts so it's the exact same diagram here but there's no prompt it's just uh
你只是在模仿人类棋手，如果你只是模仿，就无法从根本上超越人类棋手。但强化学习的过程要强大得多。在围棋中应用强化学习，意味着系统会选择那些从经验和统计角度来看能导致胜利的走法。AlphaGo 就是这样一个系统，它通过自我对弈，并利用强化学习来进行推演。这里的原理和大语言模型的强化学习是一样的，只是没有问题提示，它只是……
(2:43:10) because there's no prompt it's just a fixed game of Go but it's trying out lots of solutions it's trying out lots of plays and then the games that lead to a win instead of a specific answer are reinforced they're they're made stronger and so um the system is learning basically the sequences of actions that empirically and statistically lead to winning the game and reinforcement learning is not going to be constrained by human performance and reinforcement learning can do significantly better and overcome even the top players like Lisa
因为没有问题提示，只是固定的围棋游戏。但它尝试了很多解决方案，尝试了很多走法，然后那些能带来胜利的棋局（而不是某个特定的答案）会得到强化，变得更具优势。所以这个系统基本上是在学习那些从经验和统计角度来看能赢得比赛的行动序列。强化学习不会受限于人类的表现，它可以表现得更好，甚至超越像李世石这样的顶级棋手。
(2:43:41) Dole and so uh probably they could have run this longer and they just chose to crop it at some point because this costs money but this is very powerful demonstration of reinforcement learning and we're only starting to kind of see hints of this diagram in

larger language models for reasoning problems so we're not going to get too far by just imitating experts we need to go beyond that set up these like little game environments and get let let the system discover reasoning traces or like ways of solving problems uh that are unique

所以，他们可能本可以让训练持续更久，只是因为成本原因选择在某个点停止。但这是强化学习的一个非常有力的证明。我们才刚刚开始在大语言模型解决推理问题的过程中看到类似这种图表所展示的情况。仅仅模仿专家我们无法取得更大的突破，我们需要超越这一点，建立类似小游戏的环境，让系统发现独特的推理过程或解决问题的方法……

(2:44:14) and that uh just basically work well now on this aspect of uniqueness notice that when you're doing reinforcement learning nothing prevents you from veering off the distribution of how humans are playing the game and so when we go back to uh this alphao search here one of the suggested modifications is called move 37 and move 37 in alphao is referring to a specific point in time where alphago basically played a move that uh no human expert would play uh so the probability of this move uh to be played by a human player was evaluated

这些方法要行之有效。现在，关于这种独特性，要注意的是，在进行强化学习时，没有什么能阻止系统偏离人类的游戏方式。当我们回顾 AlphaGo 的搜索过程时，其中一个被称为“第 37 步”的走法很有名。在 AlphaGo 的对局中，第 37 步指的是它走出了一步任何人类专家都不会走的棋。据评估，人类棋手走出这步棋的概率……

(2:44:48) to be about 1 in 10th ,000 so it's a very rare move but in retrospect it was a brilliant move so alphago in the process of reinforcement learning discovered kind of like a strategy of playing that was unknown to humans and but is in retrospect uh brilliant I recommend this YouTube video um leis do versus alphao move 37 reactions and Analysis and this is kind of what it looked like when alphao played this move value that's a very that's a very surprising move I thought I thought it was I thought it was a mistake when I see this move anyway so

大约是万分之一，这是非常罕见的一步棋。但事后看来，这是一步妙棋。所以 AlphaGo 在强化学习过程中发现了一种人类未知的下棋策略，事后看来这一策略非常高明。我推荐大家看一个 YouTube 视频，叫《李世石对阵 AlphaGo：第 37 步棋的反应与分析》，这能让你了解 AlphaGo 走出这步棋时的情况。这步棋非常令人惊讶，我看到这步棋的时候，还以为是个失误呢。不管怎么说……

(2:45:25) basically people are kind of freaking out because it's a it's a move that a human would not play that alphago played because in its training uh this move seemed to be a good idea it just happens not to be a kind of thing that a humans would would do and so that is again the power of reinforcement learning and in principle we can actually see the equivalence of that if we continue scaling this Paradigm in language models and what that looks like is kind of unknown so so um what does it mean to solve problems in such a way that uh

人们对此感到震惊，因为这是人类不会走的一步棋，而 AlphaGo 走了出来。因为在它的训练过程中，这步棋似乎是个好主意，只是碰巧不是人类会采取的走法。这再次体现了强化学习的力量。原则上，如果我们在语言模型中继续扩展这种范式，我们也能看到类似的情况，不过具体会是什么样还不太清楚。那么，以一种……

(2:45:55) even humans would not be able to get how can you be better at reasoning or thinking than humans how can you go beyond just uh a thinking human like maybe it means discovering analogies that humans would not be able to uh create or maybe it's like a new thinking strategy it's kind of hard to think through uh maybe it's a holy new language that actually is not even English maybe it discovers its own language that is a lot better at thinking um because the model is unconstrained to even like stick with English uh so maybe it takes a different

连人类都无法理解的方式解决问题意味着什么呢？怎样才能比人类更擅长推理或思考呢？怎样才能超越人类的思维方式呢？也许这意味着发现人类无法创造的类比，或者是一种全新的思维策略，这很难想象。也许是一种全新的语言，甚至不是英语，也许模型会发现一种更有利于思考的自身语言。因为模型不受限于使用英语，所以也许它会采用不同的……

(2:46:27) language to think in or it discovers its own language so in principle the behavior of the system is a lot less defined it is open to do whatever works and it is open to also slowly Drift from the distribution of its training data which is English but all of that can only be done if we have a very large diverse set of problems in which the these strategy can be refined and perfected and so that is a lot of the frontier LM research that's going on right now is trying to kind of create those kinds of prompt distributions that

语言来思考，或者发现自己的语言。原则上，系统的行为具有很大的不确定性，它可以尝试任何有效的方法，甚至可以慢慢偏离其训练数据（如英语）的分布。但所有这些只有在我们拥有大量多样的问题时才有可能实现，在这些问题中，这些策略可以得到优化和完善。这就是目前很多大语言模型前沿研究正在进行的工作，试图创建各种……

(2:46:57) are large and diverse these are all kind of like game environments in which the

llms can practice their thinking and uh it's kind of like writing you know these practice problems we have to create practice problems for all of domains of knowledge and if we have practice problems and tons of them the models will be able to reinforcement learning reinforcement learn on them and kind of uh create these kinds of uh diagrams but in the domain of open thinking instead of a closed domain like game of Go there's one more section within
大规模、多样化的问题提示分布，这些就像是游戏环境，大语言模型可以在其中锻炼它们的思考能力。这有点像编写练习题，我们必须为所有知识领域创建练习题。如果我们有大量的练习题，模型就可以在上面进行强化学习，有点像…… 绘制出类似这样的图表，但这是在开放思维领域，而不是像围棋这样的封闭领域。强化学习中还有一部分内容……
(2:47:28) reinforcement learning that I wanted to cover and that is that of learning in unverifiable domains so so far all of the problems that we've looked at are in what's called verifiable domains that is any candidate solution we can score very easily against a concrete answer so for example answer is three and we can very easily score these Solutions against the answer of three either we require the models to like box in their answers and then we just check for equality of whatever is in the box with the answer or you can also use uh
我想介绍一下，那就是在不可验证领域的学习。到目前为止，我们所讨论的问题都属于可验证领域，也就是说，任何候选解决方案都可以很容易地根据一个具体答案进行评分。例如，答案是 3，我们可以很容易地根据这个答案来评估这些解决方案。我们可以要求模型把答案框起来，然后检查框里的内容是否与答案一致，或者你也可以使用……
(2:47:58) kind of what's called an llm judge so the llm judge looks at a solution and it gets the answer and just basically scores the solution for whether it's consistent with the answer or not and llms uh empirically are good enough at the current capability that they can do this fairly reliably so we can apply those kinds of techniques as well in any case we have a concrete answer and we're just checking Solutions again against it and we can do this automatically with no kind of humans in the loop the problem is that we can't apply the strategy in
一种所谓的大语言模型评判方法。大语言模型评判器会查看解决方案和答案，基本上就是判断解决方案与答案是否一致。根据经验，目前大语言模型的能力已经足够好，可以相当可靠地完成这项工作。所以在任何情况下，只要我们有具体的答案，就可以用这些技术来检查解决方案，而且整个过程可以自动完成，无需人工干预。但问题是，这种策略在……
(2:48:25) what's called unverifiable domains so usually these are for example creative writing tasks like write a joke about Pelicans or write a poem or summarize a paragraph or something like that in these kinds of domains it becomes harder to score our different solutions to this problem so for example writing a joke about Pelicans we can generate lots of different uh jokes of course that's fine for example we can go to chbt and we can get it to uh generate a joke about Pelicans uh "so much stuff in their beaks because they don't bellan in
所谓的不可验证领域并不适用。通常，这些领域涉及创造性写作任务，比如写一个关于鹈鹕的笑话、写一首诗或总结一段文字等等。在这些领域中，很难对针对同一问题生成的不同解决方案进行评分。例如，在写关于鹈鹕的笑话时，我们当然可以生成很多不同的笑话。比如，我们可以在 ChatGPT 中让它生成一个关于鹈鹕的笑话：“它们嘴里能装那么多东西，是因为它们不会把东西放在背包里（此处原句‘they don't bellan in backpacks’可能存在错误，推测是一种诙谐表达 ）”。
(2:48:57) backpacks"what okay we can uh we can try something else"why don't Pelicans ever pay for their drinks because they always B it to someone else" haha okay so these models are not obviously not very good at humor actually I think it's pretty fascinating because I think humor is secretly very difficult and the model have the capability I think anyway in any case you could imagine creating lots of jokes the problem that we are facing is how do we score them now in principle we could of course get a human to look at all
哈哈，好吧。我们还可以试试别的：“为什么鹈鹕从不为它们的饮料买单？因为它们总是把账记在别人头上”。哈哈，好的。实际上，很明显这些模型在幽默方面表现并不出色。我觉得这很有意思，因为我认为幽默其实很难把握，不过模型还是有一定能力的。不管怎样，你可以想象生成很多笑话。但我们面临的问题是，如何给它们评分呢？原则上，我们当然可以让一个人来查看所有……
(2:49:29) these jokes just like I did right now the problem with that is if you are doing reinforcement learning you're going to be doing many thousands of updates and for each update you want to be looking at say thousands of prompts and for each prompt you want to be potentially looking at looking at hundred or thousands of different kinds of generations and so there's just like way too many of these to look at and so um in principle you could have a human inspect all of them and score them and decide that okay maybe this one is funny
这些笑话，就像我刚才做的那样。但问题在于，如果你在进行强化学习，你要进行成千上万次的更新。每次更新

时，你可能要查看数千个提示，对于每个提示，你可能要查看成百上千种不同的生成结果。这样一来，需要查看的内容太多了。所以，原则上虽然可以让人工检查所有内容并评分，比如判断这个笑话可能很有趣，那个……
(2:49:56) and uh maybe this one is funny and this one is funny and we could train on them to get the model to become slightly better at jokes um in the context of pelicans at least um the problem is that it's just like way too much human time this is an unscalable strategy we need some kind of an automatic strategy for doing this and one sort of solution to this was proposed in this paper uh that introduced what's called reinforcement learning from Human feedback and so this was a paper from open at the time and many of these
这个也有趣，然后基于这些来训练模型，让它至少在关于鹈鹕的笑话方面表现得稍微好一些。但问题是，这太耗费人力时间了，这不是一个可扩展的策略。我们需要某种自动化的策略来解决这个问题。有一篇论文提出了一种解决方案，引入了所谓的人类反馈强化学习（Reinforcement Learning from Human Feedback，RLHF）。这是当时 OpenAI 发表的一篇论文，论文的很多作者……
(2:50:25) people are now um co-founders in anthropic um and this kind of proposed a approach for uh basically doing reinforcement learning in unverifiable domains so let's take a look at how that works so this is the cartoon diagram of the core ideas involved so as I mentioned the native approach is if we just set Infinity human time we could just run RL in these domains just fine so for example we can run RL as usual if I have Infinity humans I would I just want to do and these are just cartoon numbers I want to do 1,000 updates where
现在是 Anthropic 公司的联合创始人。这种方法主要是为在不可验证领域进行强化学习提供了一种途径。我们来看看它是如何工作的。这是相关核心思想的示意图。正如我提到的，传统的方法是，如果有无限的人力时间，我们可以在这些领域顺利进行强化学习。例如，我们可以像往常一样进行强化学习，如果我有无限多的人，我会（这里只是假设的数字）进行 1000 次更新，每次更新……
(2:50:58) each update will be on 1,000 prompts and in for each prompt we're going to have 1,000 roll outs that we're scoring so we can run RL with this kind of a setup the problem is in the process of doing this I will need to run one I will need to ask a human to evaluate a joke a total of 1 billion times and so that's a lot of people looking at really terrible jokes so we don't want to do that so instead we want to take the arlef approach so um in our Rel of approach we are kind of like the the core trick is that of indirection so we're going to
针对 1000 个提示，对于每个提示，我们要对 1000 个生成结果进行评分。我们可以按照这种设置进行强化学习。但问题是，在这个过程中，我总共需要让人评估 10 亿次笑话，这意味着很多人要去看大量并不好笑的笑话，我们并不想这样做。所以我们采用 RLHF 这种方法。在 RLHF 方法中，核心技巧是间接评估。我们要……
(2:51:33) involve humans just a little bit and the way we cheat is that we basically train a whole separate neural network that we call a reward model and this neural network will kind of like imitate human scores so we're going to ask humans to score um roll we're going to then imitate human scores using a neural network and this neural network will become a kind of simulator of human preferences and now that we have a neural network simulator we can do RL against it so instead of asking a real human we're asking a simulated human for
只让人类参与少量工作。具体做法是，我们训练一个完全独立的神经网络，称之为奖励模型。这个神经网络会模仿人类的评分。我们让人类对生成结果进行评分，然后用神经网络来模仿这些评分，这个神经网络就会成为人类偏好的模拟器。现在有了这个神经网络模拟器，我们就可以基于它进行强化学习。所以，我们不再询问真实的人类，而是询问模拟的“人类”来……
(2:52:06) their score of a joke as an example and so once we have a simulator we're often racist because we can query it as many times as we want to and it's all whole automatic process and we can now do reinforcement learning with respect to the simulator and the simulator as you might expect is not going to be a perfect human but if it's at least statistically similar to human judgment then you might expect that this will do something and in practice indeed uh it does so once we have a simulator we can do RL and everything works great so let
对一个笑话进行评分，举个例子。一旦有了这个模拟器，我们就方便多了，因为我们可以随意多次查询它，而且这完全是自动化过程。现在我们可以基于这个模拟器进行强化学习。正如你可能预料的，这个模拟器并不完全等同于真实的人类，但如果它在统计上至少与人类的判断相似，那么你可能会期望它能起到一定作用。实际上，确实如此。一旦有了模拟器，我们就可以进行强化学习，一切似乎都很顺利。那么我们来……
(2:52:35) me show you a cartoon diagram a little bit of what this process looks like although the details are not 100 like super important it's just a core idea of how this works so here I have a cartoon diagram of a hypothetical example of what training the reward model would look like so we have a prompt like "write a joke about picans" and then here we have five separate roll outs so these are all five different jokes just like this

one now the first thing we're going to do is we are going to ask a human to uh order these jokes from the best to

给你展示一下这个过程的示意图，虽然具体细节不是非常重要，关键是理解它的核心工作原理。这里有一个假设的例子，展示训练奖励模型的过程。我们有一个提示，比如“写一个关于鹈鹕的笑话”，然后有五个不同的生成结果，就像这样。我们要做的第一件事是让一个人将这些笑话从最好到最差进行排序。

(2:53:05) worst so this is uh so here this human thought that this joke is the best the funniest so number one joke this is number two joke number three joke four and five so this is the worst joke we're asking humans to order instead of give scores directly because it's a bit of an easier task it's easier for a human to give an ordering than to give precise scores now that is now the supervision for the model so the human has ordered them and that is kind of like their contribution to the training process but now separately what we're

所以，这个人认为这个笑话是最好、最有趣的，所以把它排在第一位，这个是第二个，第三个，第四个，第五个，这个是最差的笑话。我们让人类进行排序而不是直接打分，因为这是一项相对容易的任务，对人类来说，排序比给出精确分数更容易。现在，这就是对模型的监督信息。人类完成了排序，这算是他们对训练过程的贡献。但现在，我们还要单独做一件事……

(2:53:36) going to do is we're going to ask a reward model uh about its scoring of these jokes now the reward model is a whole separate neural network completely separate neural net um and it's also probably a transform uh but it's not a language model in the sense that it generates diverse language Etc it's just a scoring model so the reward model will take as an input The Prompt number one and number two a candidate joke so um those are the two inputs that go into the reward model so here for example the reward model would

我们要让奖励模型对这些笑话进行评分。奖励模型是一个完全独立的神经网络，它可能也是基于 Transformer 架构，但它不是那种会生成多样语言的语言模型，它只是一个评分模型。奖励模型会将提示和候选笑话作为输入，这就是输入到奖励模型的两个内容。例如，在这里，奖励模型会……

(2:54:09) be taken this prompt and this joke now the output of a reward model is a single number and this number is thought of as a score and it can range for example from Z to one so zero would be the worst score and one would be the best score so here are some examples of what a hypothetical reward model at some stage in the training process would give uh s scoring to these jokes so 0.

以这个提示和这个笑话作为输入。奖励模型的输出是一个单一的数字，这个数字被视为分数，例如，分数范围可以从 0 到 1，0 表示最差的分数，1 表示最好的分数。这里是在训练过程中某个阶段，一个假设的奖励模型对这些笑话评分的示例。比如，0.

(2:54:33) 1 is a very low score 08 is a really high score and so on and so now um we compare the scores given by the reward model with uh the ordering given by the human and there's a precise mathematical way to actually calculate this uh basically set up a loss function and calculate a kind of like a correspondence here and uh update a model based on it but I just want to give you the intuition which is that as an example here for this second joke the the human thought that it was the funniest and the model kind of agreed right 08 is a relatively high

1 是一个很低的分数，0.8 是一个很高的分数，等等。现在，我们将奖励模型给出的分数与人类的排序进行比较。有一个精确的数学方法来进行这个计算，基本上就是设置一个损失函数，计算两者之间的某种对应关系，并据此更新模型。但我只是想让你有个直观的理解，例如，对于第二个笑话，人类认为它是最有趣的，模型也比较认同，对吧？0.8 是一个相对较高的分数……

(2:55:06) score but this score should have been even higher right so after an update we would expect that maybe this score should have been will actually grow after an update of the network to be like say 081 or something um for this one here they actually are in a massive disagreement because the human thought that this was number two but here the the score is only 0.

但这个分数其实应该更高，对吧？所以在一次更新之后，我们会期望这个分数在网络更新后可能会增长，比如增长到 0.81 之类的。对于这个笑话，人类和模型的看法存在很大分歧，因为人类认为它排第二，但模型给出的分数只有 0.

(2:55:28) 1 and so this score needs to be much higher so after an update on top of this um kind of a supervision this might grow a lot more like maybe it's 0.15 or something like that um and then here the human thought that this one was the worst joke but here the model actually gave it a fairly High number so you might expect that after the update uh this would come down to maybe 3 3.

1，所以这个分数需要大幅提高。在基于这种监督进行更新后，它可能会大幅增长，比如增长到 0.15 左右。然后，对于这个笑话，人类认为它是最差的，但模型实际上给了它一个相当高的分数。所以你可能会期望在更新之后，这个分数可能会下降到，比如说，0.35 左右。

(2:55:50) 5 or something like that so basically we're doing what we did before we're slightly nudging the predictions from the models using a neural network training process and we're trying to make the reward model scores be consistent with human ordering and so um as we update the reward model on human data it becomes better and better simulator of the scores and orders uh that humans provide and then becomes kind of like the the neural the simulator of human preferences which we can then do RL against but critically we're not asking humans one billion times to look at a

所以基本上，我们做的和之前一样，通过神经网络训练过程对模型的预测进行微调，试图让奖励模型的分数与人类的排序一致。所以，随着我们用人类数据更新奖励模型，它会越来越能模拟人类给出的分数和排序，进而成为人类偏好的神经网络模拟器。这样我们就可以基于它进行强化学习。关键是，我们不用让人类去看 10 亿次笑话，而是……

(2:56:25) joke we're maybe looking at th000 prompts and five roll outs each so maybe 5,000 jokes that humans have to look at in total and they just give the ordering and then we're training the model to be consistent with that ordering and I'm skipping over the mathematical details but I just want you to understand a high level idea that uh this reward model is do is basically giving us this scour and we have a way of training it to be consistent with human orderings and that's how rhf works okay so that is the rough idea we basically train

可能只让人类查看 1000 个提示，每个提示对应 5 个生成结果，所以人类总共大概只需要查看 5000 个笑话并给出排序。然后我们训练模型，使其与这些排序一致。我略过了数学细节，只是想让你理解一个大致的概念：这个奖励模型基本上就是给我们提供分数，并且我们有办法训练它，使其与人类的排序保持一致，这就是 RLHF 的工作原理。好的，这就是大致的思路，我们基本上是训练……

(2:57:25) simulators of humans and RL with respect to those simulators now I want to talk about first the upside of reinforcement learning from Human feedback the first thing is that this allows us to run reinforcement learning which we know is incredibly powerful kind of set of techniques and it allows us to do it in arbitrary domains and including the ones that are unverifiable so things like summarization and poem writing joke writing or any other creative writing really uh in domains outside of math and code Etc now empirically what we see when we

人类偏好的模拟器，并基于这些模拟器进行强化学习。现在我想先谈谈人类反馈强化学习的优点。首先，它让我们能够使用强化学习，我们知道强化学习是一组非常强大的技术，它使我们能够在任意领域，包括不可验证的领域进行强化学习。比如总结、写诗、写笑话或其他任何创造性写作，在数学和代码等领域之外的领域都适用。现在，从经验上看，当我们……

(2:57:53) actually apply rhf is that this is a way to improve the performance of the model and uh I have a top answer for why that might be but I don't actually know that it is like super well established on like why this is you can empirically observe that when you do rhf correctly the models you get are just like a little bit better um but as to why is I think like not as clear so here's my best guess my best guess is that this is possibly mostly due to the discriminator generator Gap what that means is that in many

实际应用 RLHF 时，我们发现这是一种提高模型性能的方法。我有一个关于为什么会这样的猜测，但我不确定这个解释是否已经得到充分验证。你可以从经验上观察到，当正确应用 RLHF 时，得到的模型确实会有一些提升。但至于具体原因，我觉得还不是很清楚。我最好的猜测是，这可能主要是由于判别器和生成器之间的差距。这意味着在很多……

(2:58:26) cases it is significantly easier to discriminate than to generate for humans so in particular an example of this is um in when we do supervised fine-tuning right sft we're asking humans to generate the ideal assistant response and in many cases here um as I've shown it uh the ideal response is very simple to write but in many cases might not be so for example in summarization or poem writing or joke writing like how are you as a human assist as a human labeler um supposed to give the ideal response in these cases it requires creative human

情况下，对人类来说，判别比生成要容易得多。具体的一个例子是，当我们进行监督微调（SFT）时，我们让人类生成理想的助手回复。在很多情况下，就像我展示过的，理想回复很容易写，但在很多其他情况下可能并非如此。比如在总结、写诗或写笑话时，作为人类标注员，你应该如何给出理想的回复呢？在这些情况下，需要人类进行创造性写作……

(2:59:33) writing to do that and so rhf kind of sidesteps this because we get um we get to ask people a significantly easier question as a data labelers they're not asked to write poems directly they're just given five poems from the model and they're just asked to order them and so that's just a much easier task for a human labeler to do and so what I think this allows you to do basically is it um it kind of like allows a lot more higher accuracy data because we're not asking people to do the generation task which can be extremely difficult like we're

才能完成。而 RLHF 在一定程度上避开了这个问题，因为作为数据标注员，我们让人们回答一个容易得多的问题。

他们不用直接写诗，只是拿到模型生成的五首诗，然后对它们进行排序。所以这对人类标注员来说是一项容易得多的任务。所以我认为这样做基本上可以获取更多更高质量的数据，因为我们没有让人们去完成可能极其困难的生成任务，比如……

(3:00:06) not asking them to do creative writing we're just trying to get them to distinguish between creative writings and uh find the ones that are best and that is the signal that humans are providing just the ordering and that is their input into the system and then the system in rhf just discovers the kinds of responses that would be graded well by humans and so that step of indirection allows the models to become a bit better so that is the upside of our LF it allows us to run RL it empirically results in better models and

我们没有让他们进行创造性写作，只是让他们区分不同的创造性作品，并找出最好的那些。这就是人类提供的信号，仅仅是排序信息，这就是他们输入到系统中的内容。然后在 RLHF 中，系统会发现那些能得到人类较高评价的回复。所以这种间接的方式让模型表现得更好一些。这就是 RLHF 的优点，它让我们能够进行强化学习，从经验上看能得到更好的模型，并且……

(3:00:38) it allows uh people to contribute their supervision uh even without having to do extremely difficult tasks um in the case of writing ideal responses unfortunately our HF also comes with significant downsides and so um the main one is that basically we are doing reinforcement learning not with respect to humans and actual human judgment but with respect to a lossy simulation of humans right and this lossy simulation could be misleading because it's just a it's just a simulation right it's just a language model that's kind of outputting scores

它使得人们即便无需执行极其困难的任务，也能够贡献他们的监督。就撰写理想回复而言，不幸的是，我们的人类反馈（HF）也存在显著的缺点。主要的一点是，基本上我们所进行的强化学习，并非基于真实的人类和人类的实际判断，而是基于对人类的一种有损耗的模拟，对吧？而这种有损耗的模拟可能会产生误导，因为它仅仅是一种模拟，只是一个会输出分数的语言模型。

(3:01:09) and it might not perfectly reflect the opinion of an actual human with an actual brain in all the possible different cases so that's number one which is actually something even more subtle and devious going on that uh really dramatically holds back our LF as a technique that we can really scale to significantly um kind of Smart Systems and that is that reinforcement learning is extremely good at discovering a way to game the model to game the simulation so this reward model that we're constructing here that gives the course

而且在所有可能的不同情况下，它可能无法完美反映一个拥有真实大脑的真实人类的意见。所以这是第一点，实际上这里还存在一些更为微妙和棘手的情况，这极大地限制了我们将人类反馈（HF）作为一种能够真正大规模应用于智能系统的技术。也就是说，强化学习非常擅长找到一种方法来“玩弄”模型，即“玩弄”这种模拟。所以我们在此构建的这个给出分数的奖励模型……

(3:01:43) these models are Transformers these Transformers are massive neurals they have billions of parameters and they imitate humans but they do so in a kind of like a simulation way now the problem is that these are massive complicated systems right there's a billion parameters here that are outputting a single score it turns out that there are ways to gain these models you can find kinds of inputs that were not part of their training set and these inputs inexplicably get very high scores but in a fake way so very often what you find

这些模型是 Transformer 模型，这些 Transformer 模型是大规模的神经网络，它们拥有数十亿个参数，并且模仿人类，但它们是以一种模拟的方式来进行模仿。现在的问题是，这些是大规模且复杂的系统，对吧？这里有数十亿个参数输出一个单一的分数。事实证明，存在一些方法可以“玩弄”这些模型。你可以找到一些并非它们训练集一部分的输入类型，而这些输入会莫名其妙地得到非常高的分数，但却是以一种虚假的方式。所以，你经常会发现……

(3:02:17) if you run our lch for very long so for example if we do 1,000 updates which is like say a lot of updates you might expect that your jokes are getting better and that you're getting like real bangers about Pelicans but that's not EXA exactly what happens what happens is that uh in the first few hundred steps the jokes about Pelicans are probably improving a little bit and then they actually dramatically fall off the cliff and you start to get extremely nonsensical results like for example you start to get um the top joke about

(3:02:45) Pelicans starts to be the and this makes no sense right like when you look at it why should this be a top joke but when you take the the and you plug it into your reward model you'd expect score of zero but actually the reward model loves this as a joke it will tell you that the the the theth is a score of 1.

如果你长时间运行我们的语言模型训练过程（比如进行 1000 次更新，这算是很多次更新了），你可能会期望你的笑话变得更好，比如得到一些关于鹈鹕的非常棒的笑话，但实际情况并非完全如此。实际发生的是，在前几百步中，关于鹈鹕的笑话可能会有一点改进，然后它们实际上会急剧下降，你开始得到极其荒谬的结果。例如，你开始

得到…… 关于鹈鹕的最佳笑话开始变成这样（某条毫无意义的内容），这毫无道理，对吧？就像当你看到它时，会想为什么这会是一个最佳笑话呢？但是当你把它输入到你的奖励模型中时，你期望得到的分数是零，但实际上奖励模型却非常喜欢把它当作一个笑话，它会告诉你这条内容的分数是 1。
(3:03:06) Z this is a top joke and this makes no sense right but it's because these models are just simulations of humans and they're massive neural lots and you can find inputs at the bottom that kind of like get into the part of the input space that kind of gives you nonsensical results these examples are what's called adversarial examples and I'm not going to go into the topic too much but these are adversarial inputs to the model they are specific little inputs that kind of go between the nooks and crannies of the model and give nonsensical results at
这是一个最佳笑话，这毫无道理，对吧？但这是因为这些模型只是对人类的模拟，而且它们是大规模的神经网络，你可以找到一些输入，这些输入会进入到输入空间的某些部分，从而产生荒谬的结果。这些例子被称为对抗性示例，我不会过多深入这个话题，但这些是模型的对抗性输入。它们是一些特定的小输入，会在模型的各个角落之间钻空子，并在顶部产生荒谬的结果。
(3:03:33) the top now here's what you might imagine doing you say okay the the the is obviously not score of one um it's obviously a low score so let's take the the the the the let's add it to the data set and give it an ordering that is extremely bad like a score of five and indeed your model will learn that the D should have a very low score and it will give it score of zero the problem is that there will always be basically infinite number of nonsensical adversarial examples hiding in the model if you iterate this process many times
现在，这是你可能会想到要做的事。你会说，“好吧，（这条内容）显然不该得 1 分，它显然是个低分。所以，咱们就拿（这条内容）来说，把它添加到数据集中，然后给它一个极差的排序，比如 5 分。” 确实，你的模型会学到（这条内容）应该得一个非常低的分数，并且它会给它打出 0 分。问题是，如果你多次重复这个过程，基本上在模型中总会隐藏着无穷无尽的无意义的对抗性示例。
(3:04:02) and you keep adding nonsensical stuff to your reward model and giving it very low scores you can you'll never win the game uh you can do this many many rounds and reinforcement learning if you run it long enough will always find a way to gain the model it will discover adversarial examples it will get get really high scores uh with nonsensical results and fundamentally this is because our scoring function is a giant neural nut and RL is extremely good at finding just the ways to trick it uh so long story short you always run rhf put
(3:04:37) for maybe a few hundred updates the model is getting better and then you have to crop it and you are done you can't run too much against this reward model because the optimization will start to game it and you basically crop it and you call it and you ship it um and uh you can improve the reward model but you kind of like come across these situations eventually at some point so rhf basically what I usually say is that RF is not RL and what I mean by that is I mean RF is RL obviously but it's not RL in the magical sense this is not RL
(3:05:12) that you can run indefinitely these kinds of problems like where you are getting con correct answer you cannot gain this as easily you either got the correct answer or you didn't and the scoring function is much much simpler you're just looking at the boxed area and seeing if the result is correct so it's very difficult to gain these functions but uh gaming a reward model is possible now in these verifiable domains you can run RL indefinitely you could run for tens of thousands hundreds of thousands of steps
然后你不断地往奖励模型中添加无意义的内容，并给它们很低的分数，可你永远都无法在这场 “游戏” 中取胜。你可以进行很多很多轮这样的操作，而强化学习如果运行的时间足够长，总是会找到一种 “玩弄” 模型的方法。它会发现对抗性示例，会得出毫无意义的结果却获得非常高的分数。从根本上来说，这是因为我们的评分函数是一个庞大的神经网络，而强化学习极其擅长找到欺骗它的方法。
所以长话短说，当你运行基于人类反馈的强化学习（RLHF）时，也许在前几百次更新中，模型会有所改进，然后你就必须停止训练。因为如果你针对这个奖励模型运行太多次，优化过程就会开始 “玩弄” 模型，所以基本上你要停止训练，结束这一过程并发布模型。你可以改进奖励模型，但在某个时候你最终还是会遇到这样的情况。
所以基本上，我常说基于人类反馈的强化学习（RLHF）不是真正意义上的强化学习（RL）。我的意思是，RLHF 显然属于强化学习（RL）的范畴，但它不是那种具有神奇效果的强化学习。这不是那种你可以无限制运行的强化学习。像在那些你要得出正确答案的问题中，你没办法那么轻易地 “玩弄” 它。你要么得到正确答案，要么得不到，而且评分函数要简单得多，你只是查看特定区域，看看结果是否正确，所以很难 “玩弄” 这些函数。
但在现在这种情况下，“玩弄” 一个奖励模型是有可能的。在这些可验证的领域中，你可以无限制地运行强化学习，你可以运行数万次、甚至数十万次。
(3:05:40) and discover all kinds of really crazy strategies that we might not even ever think about of Performing really well for all these problems in the game of Go there's no way to to beat to basically game uh the winning of a game or the losing of a game we have

a perfect simulator we know all the different uh where all the stones are placed and we can calculate uh whether someone has won or not there's no way to gain that and so you can do RL indefinitely and you can eventually be beat even leol but with models like this which are gameable
(3:06:12) you cannot repeat this process indefinitely so I kind of see rhf as not real RL because the reward function is gameable so it's kind of more like in the realm of like little fine-tuning it's a little it's a little Improvement but it's not something that is fundamentally set up correctly where you can insert more compute run for longer and get much better and magical results so it's it's uh it's not RL in that sense it's not RL in the sense that it lacks magic um it can find you in your model and get a better performance and
(3:06:44) indeed if we go back to chat GPT the GPT 40 model has gone through rhf because it works well but it's just not RL in the same sense rlf is like a little fine tune that slightly improves your model is maybe like the way I would think about it okay so that's most of the technical content that I wanted to cover I took you through the three major stages and paradigms of training these models pre-training supervised fine tuning and reinforcement learning and I showed you that they Loosely correspond to the process we already use for
(3:07:13) teaching children and so in particular we talked about pre-training being sort of like the basic knowledge acquisition of reading Exposition supervised fine tuning being the process of looking at lots and lots of worked examples and imitating experts and practice problems the only difference is that we now have to effectively write textbooks for llms and AIS across all the disciplines of human knowledge and also in all the cases where we actually would like them to work like code and math and you know basically all the other disciplines so
(3:07:44) we're in the process of writing textbooks for them refining all the algorithms that I've presented on the high level and then of course doing a really really good job at the execution of training these models at scale and efficiently so in particular I didn't go into too many details but these are extremely large and complicated distributed uh sort of um jobs that have to run over tens of thousands or even hundreds of thousands of gpus and the engineering that goes into this is really at the stateof the art of what's possible with computers at
(3:08:14) that scale so I didn't cover that aspect too much but um this is very kind of serious and they were underlying all these very simple algorithms ultimately now I also talked about sort of like the theory of mind a little bit of these models and the thing I want you to take away is that these models are really good but they're extremely useful as tools for your work you shouldn't uh sort of trust them fully and I showed you some examples of that even though we have mitigations for hallucinations the models are not perfect and they will
(3:08:45) hallucinate still it's gotten better over time and it will continue to get better but they can hallucinate in other words in in addition to that I covered kind of like what I call the Swiss cheese uh sort of model of llm capabilities that you should have in your mind the models are incredibly good across so many different disciplines but then fail randomly almost in some unique cases so for example what is bigger 9.11 or 9.
(3:09:08) 9 like the model doesn't know but simultaneously it can turn around and solve Olympiad questions and so this is a hole in the Swiss cheese and there are many of them and you don't want to trip over them so don't um treat these models as infallible models check their work use them as tools use them for inspiration use them for the first draft but uh work with them as tools and be ultimately respons responsible for the you know product of your work and that's roughly what I wanted to talk about this is how they're trained
(3:09:41) and this is what they are let's now turn to what are some of the future capabilities of these models uh probably what's coming down the pipe and also where can you find these models I have a few blow points on some of the things that you can expect coming down the pipe the first thing you'll notice is that the models will very rapidly become multimodal everything I talked about above concerned text but very soon we'll have llms that can not just handle text but they can also operate natively and very easily over audio so they can hear
(3:10:08) and speak and also images so they can see and paint and we're already seeing the beginnings of all of this uh but this will be all done natively inside inside the language model and this will enable kind of like natural conversations and roughly speaking the reason that this is actually no different from everything we've covered above is that as a baseline you can tokenize audio and images and apply the exact same approaches of everything that we've talked about above so it's not a fundamental change it's just uh it's
(3:10:36) just a to we have to add some tokens so as an example for tokenizing audio we

can look at slices of the spectrogram of the audio signal and we can tokenize that and just add more tokens that suddenly represent audio and just add them into the context windows and train on them just like above the same for images we can use patches and we can separately tokenize patches and then what is an image an image is just a sequence of tokens and this actually kind of works and there's a lot of early work in this direction and so we can
(3:11:07) just create streams of tokens that are representing audio images as well as text and interpers them and handle them all simultaneously in a single model so that's one example of multimodality uh second something that people are very interested in is currently most of the work is that we're handing individual tasks to the models on kind of like a silver platter like please solve this task for me and the model sort of like does this little task but it's up to us to still sort of like organize a coherent execution of
(3:11:35) tasks to perform jobs and the models are not yet at the capability required to do this in a coherent error correcting way over long periods of time so they're not able to fully string together tasks to perform these longer running jobs but they're getting there and this is improving uh over time but uh probably what's going to happen here is we're going to start to see what's called agents which perform tasks over time and you you supervise them and you watch their work and they come up to once in a while report progress and so on so we're
(3:12:07) going to see more long running agents uh tasks that don't just take you know a few seconds of response but many tens of seconds or even minutes or hours over time uh but these uh models are not infallible as we talked about above so all of this will require supervision so for example in factories people talk about the human to robot ratio uh for automation I think we're going to see something similar in the digital space where we are going to be talking about human to agent ratios where humans becomes a lot more supervisors of agent
(3:12:36) tasks um in the digital domain uh next um I think everything is going to become a lot more pervasive and invisible so it's kind of like integrated into the tools and everywhere um and in addition kind of like computer using so right now these models aren't able to take actions on your behalf but I think this is a separate bullet point um if you saw chpt launch the operator then uh that's one early example of that where you can actually hand off control to the model to perform you know keyboard and mouse actions on your
(3:13:10) behalf so that's also something that that I think is very interesting the last point I have here is just a general comment that there's still a lot of research to potentially do in this domain main one example of that uh is something along the lines of test time training so remember that everything we've done above and that we talked about has two major stages there's first the training stage where we tune the parameters of the model to perform the tasks well once we get the parameters we fix them and then we deploy the model
(3:13:35) for inference from there the model is fixed it doesn't change anymore it doesn't learn from all the stuff that it's doing a test time it's a fixed um number of parameters and the only thing that is changing is now the token inside the context windows and so the only type of learning or test time learning that the model has access to is the in context learning of its uh kind of like uh dynamically adjustable context window depending on like what it's doing at test time so but I think this is still different from humans who actually are
(3:14:05) able to like actually learn uh depending on what they're doing especially when you sleep for example like your brain is updating your parameters or something like that right so there's no kind of equivalent of that currently in these models and tools so there's a lot of like um more wonky ideas I think that are to be explored still and uh in particular I think this will be necessary because the context window is a finite and precious resource and especially once we start to tackle very long running multimodal tasks and we're
(3:14:30) putting in videos and these token windows will basically start to grow extremely large like not thousands or even hundreds of thousands but significantly beyond that and the only trick uh the only kind of trick we have Avail to us right now is to make the context Windows longer but I think that that approach by itself will will not will not scale to actual long running tasks that are multimodal over time and so I think new ideas are needed in some of those disciplines um in some of those kind of cases in the main where these
(3:14:58) tasks are going to require very long contexts so those are some examples of some of the things you can um expect coming down the pipe let's now turn to where you can actually uh kind of keep track of this progress and um you know be up to date with the latest and grest of what's happening in the field so I would say the three resources that I have consistently used to stay up to date are number one El Marina uh so let me show you

El Marina this is basically an llm leader board and it ranks all the top models
(3:15:30) and the ranking is based on human comparisons so humans prompt these models and they get to judge which one gives a better answer they don't know which model is which they're just looking at which model is the better answer and you can calculate a ranking and then you get some results and so what you can hear is what you can see here is the different organizations like Google Gemini for example that produce these models when you click on any one of these it takes you to the place where that model is hosted and then here we see Google is
(3:15:57) currently on top with open AI right behind here we see deep seek in position number three now the reason this is a big deal is the last column here you see license deep seek is an MIT license model it's open weights anyone can use these weights uh anyone can download them anyone can host their own version of Deep seek and they can use it in what whatever way they like and so it's not a proprietary model that you don't have access to it's it's basically an open weight release and so this is kind of unprecedented that a model this strong
(3:16:27) was released with open weights so pretty cool from the team next up we have a few more models from Google and open Ai and then when you continue to scroll down you start to see some other Usual Suspects so xai here anthropic with son it uh here at number 14 and um then meta with llama over here so llama similar to deep seek is an open weights model and so uh but it's down here as opposed to up here now I will say that this leaderboard was really good for a long time I do think that in the last few months it's become a little bit
(3:17:05) gamed um and I don't trust it as much as I used to I think um just empirically I feel like a lot of people for example are using a Sonet from anthropic and that it's a really good model so but that's all the way down here um in number 14 and conversely I think not as many people are using Gemini but it's racking really really high uh so I think use this as a first pass uh but uh sort of try out a few of the models for your tasks and see which one performs better the second thing that I would point to is the uh AI news uh newsletter so AI
(3:17:41) news is not very creatively named but it is a very good newsletter produced by swix and friends so thank you for maintaining it and it's been very helpful to me because it is extremely comprehensive so if you go to archives uh you see that it's produced almost every other day and um it is very comprehensive and some of it is written by humans and curated by humans but a lot of it is constructed automatically with llms so you'll see that these are very comprehensive and you're probably not missing anything major if you go through it of course
(3:18:09) you're probably not going to go through it because it's so long but I do think that these summaries all the way up top are quite good and I think have some human oversight uh so this has been very helpful to me and the last thing I would point to is just X and Twitter uh a lot of um AI happens on X and so I would just follow people who you like and trust and get all your latest and greatest uh on X as well so those are the major places that have worked for me over time and finally a few words on where you can find the models and where
(3:18:38) can you use them so the first one I would say is for any of the biggest proprietary models you just have to go to the website of that LM provider so for example for open a that's uh chat I believe actually works now uh so that's for open AI now for or you know for um for Gemini I think it's gem. google.
(3:18:58) com or AI Studio I think they have two for some reason that I don't fly understand no one does um for the open weights models like deep SE CL Etc you have to go to some kind of an inference provider of LMS so my favorite one is together together. a and I showed you that when you go to the playground of together. a then you can sort of pick lots of different models and all of these are open models of different types and you can talk to them here as an example um now if you'd like to use a base model like um you know a base model
(3:19:28) then this is where I think it's not as common to find base models even on these inference providers they are all targeting assistants and chat and so I think even here I can't I couldn't see base models here so for base models I usually go to hyperbolic because they serve my llama 3.1 base and I love that model and you can just talk to it here so as far as I know this is this is a good place for a base model and I wish more people hosted base models because they are useful and interesting to work with in some cases finally you can also
(3:19:58) take some of the models that are smaller and you can run them locally and so for example deep seek the biggest model you're not going to be able to run locally on your MacBook but there are smaller versions of the deep seek model that are what's called distilled and then also you can run these models at smaller Precision so not at the native Precision of for example fp8 on deep seek or you know bf16 llama but much much lower than

that um and don't worry if you don't fully understand those details but you can run smaller versions
(3:20:27) that have been distilled and then at even lower precision and then you can fit them on your uh computer and so you can actually run pretty okay models on your laptop and my favorite I think place I go to usually is LM studio uh which is basically an app you can get and I think it kind of actually looks really ugly and it's I don't like that it shows you all these models that are basically not that useful like everyone just wants to run deep seek so I don't know why they give you these 500 different types of models they're really
(3:20:53) complicated to search for and you have to choose different distillations and different uh precisions and it's all really confusing but once you actually understand how it works and that's a whole separate video then you can actually load up a model like here I loaded up a llama 3 uh2 instruct 1 billion and um you can just talk to it so I ask for Pelican jokes and I can ask for another one and it gives me another one Etc all of this that happens here is locally on your computer so we're not actually going to anywhere anyone else
(3:21:22) this is running on the GPU on the MacBook Pro so that's very nice and you can then eject the model when you're done and that frees up the ram so LM studio is probably like my favorite one even though I don't I think it's got a lot of uiux issues and it's really geared towards uh professionals almost uh but if you watch some videos on YouTube I think you can figure out how to how to use this interface uh so those are a few words on where to find them so let me now loop back around to where we started the question was when we go to chashi
(3:21:51) pta.com and we enter some kind of a query and we hit go what exactly is happening here what are we seeing what are we talking to how does this work and I hope that this video gave you some appreciation for some of the under the hood details of how these models are trained and what this is that is coming back so in particular we now know that your query is taken and is first chopped up into tokens so we go to to tick tokenizer and here where is the place in the in the um sort of format that is for the user query we basically put in our
(3:22:28) query right there so our query goes into what we discussed here is the conversation protocol format which is this way that we maintain conversation objects so this gets inserted there and then this whole thing ends up being just a token sequence a onedimensional token sequence under the hood so Chachi PT saw this token sequence and then when we hit go it basically continues appending tokens into this list it continues the sequence it acts like a token autocomplete so in particular it gave us this response so we can basically just
(3:23:00) put it here and we see the tokens that it continued uh these are the tokens that it continued with roughly now the question becomes okay why are these the tokens that the model responded with what are these tokens where are they coming from uh what are we talking to and how do we program this system and so that's where we shifted gears and we talked about the under thehood pieces of it so the first stage of this process and there are three stages is the pre-training stage which fundamentally has to do with just
(3:23:29) knowledge acquisition from the internet into the parameters of this neural network and so the neural net internalizes a lot of Knowledge from the internet but where the personality really comes in is in the process of supervised fine-tuning here and so what what happens here is that basically the a company like openai will curate a large data set of conversations like say 1 million conversation across very diverse topics and there will be conversations between a human and an assistant and even though there's a lot
(3:23:59) of synthetic data generation used throughout this entire process and a lot of llm help and so on fundamentally this is a human data curation task with lots of humans involved and in particular these humans are data labelers hired by open AI who are given labeling instructions that they learn and they task is to create ideal assistant responses for any arbitrary prompts so they are teaching the neural network by example how to respond to prompts so what is the way to think about what came back here like what is
(3:24:33) this well I think the right way to think about it is that this is the neural network simulation of a data labeler at openai so it's as if I gave this query to a data Li open and this data labeler first reads all of the labeling instructions from open Ai and then spends 2 hours writing up the ideal assistant response to this query and uh giving it to me now we're not actually doing that right because we didn't wait two hours so what we're getting here is a neural network simulation of that process and we have to keep in mind that
(3:25:08) these neural networks don't function like human brains do they are different what's easy or hard for them is different from what's easy or hard for humans and so we really are just getting a simulation so here I shown you this is a token stream and this is fundamentally the neural network with a bunch of activations and neurons in between

this is a fixed mathematical expression that mixes inputs from tokens with parameters of the model and they get mixed up and get you the next token in a sequence but this is a finite amount of compute that
(3:25:39) happens for every single token and so this is some kind of a lossy simulation of a human that is kind of like restricted in this way and so whatever the humans write the language model is kind of imitating on this token level with only this this specific computation for every single token and sequence we also saw that as a result of this and the cognitive differences the models will suffer in a variety of ways and uh you have to be very careful with their use so for example we saw that they will suffer from hallucinations and
(3:26:14) they also we have the sense of a Swiss model of the LM capabilities where basically there's like holes in the cheese sometimes the models will just arbitrarily like do something dumb uh so even though they're doing lots of magical stuff sometimes they just can't so maybe you're not giving them enough tokens to think and maybe they're going to just make stuff up because they're mental arithmetic breaks uh maybe they are suddenly unable to count number of letters um or maybe they're unable to tell you that 911 9.11 is smaller than
(3:26:44) 9.9 and it looks kind of dumb and so so it's a Swiss cheese capability and we have to be careful with that and we saw the reasons for that but fundamentally this is how we think of what came back it's again a simulation of this neural network of a human data labeler following the labeling instructions at open a so that's what we're getting back now I do think that the uh things change a little bit when you actually go and reach for one of the thinking models like o03 mini and the reason for that is that GPT 40 basically doesn't do reinforcement
(3:27:23) learning it does do rhf but I've told you that rhf is not RL there's no there's no uh time for magic in there it's just a little bit of a fine-tuning is the way to look at it but these thinking models they do use RL so they go through this third state stage of perfecting their thinking process and discovering new thinking strategies and uh solutions to problem solving that look a little bit like your internal monologue in your head and they practice that on a large collection of practice problems that companies like openi create and
(3:27:57) curate and um then make available to the LMS so when I come here and I talked to a thinking model and I put in this question what we're seeing here is not anymore just the straightforward simulation of a human data labeler like this is actually kind of new unique and interesting um and of course open is not showing us the under thehood thinking and the chains of thought that are underlying the reasoning here but we know that such a thing exists and this is a summary of it and what we're getting here is actually not just an
(3:28:28) imitation of a human data labeler it's actually something that is kind of new and interesting and exciting in the sense that it is a function of thinking that was emergent in a simulation it's not just imitating human data labeler it comes from this reinforcement learning process and so here we're of course not giving it a chance to shine because this is not a mathematical or a reasoning problem this is just some kind of a sort of creative writing problem roughly speaking and I think it's um it's a a question an open question as to whether
(3:28:58) the thinking strategies that are developed inside verifiable domains transfer and are generalizable to other domains that are unverifiable such as create writing the extent to which that transfer happens is unknown in the field I would say so we're not sure if we are able to do RL on everything that is very verifiable and see the benefits of that on things that are unverifiable like this prompt so that's an open question the other thing that's interesting is that this reinforcement learning here is still like way too new primordial and
(3:29:29) nent so we're just seeing like the beginnings of the hints of greatness uh in the reasoning problems we're seeing something that is in principle capable of something like the equivalent of move 37 but not in the game of Go but in open domain thinking and problem solving in principle this Paradigm is capable of doing something really cool new and exciting something even that no human has thought of before in principle these models are capable of analogies no human has had so I think it's incredibly exciting that these models exist but
(3:30:01) again it's very early and these are primordial models for now um and they will mostly shine in domains that are verifiable like math en code Etc so very interesting to play with and think about and use and then that's roughly it um um I would say those are the broad Strokes of what's available right now I will say that overall it is an extremely exciting time to be in the field personally I use these models all the time daily uh tens or hundreds of times because they dramatically accelerate my work I think a lot of
(3:30:32) people see the same thing I think we're going to see a huge amount of wealth

creation as a result of these models be aware of some of their shortcomings even with RL models they're going to suffer from some of these use it as a tool in a toolbox don't trust it fully because they will randomly do dumb things they will randomly hallucinate they will randomly skip over some mental arithmetic and not get it right um they randomly can't count or something like that so use them as tools in the toolbox check their work and own the product of
(3:31:01) your work but use them for inspiration for first draft uh ask them questions but always check and verify and you will be very successful in your work if you do so uh so I hope this video was useful and interesting to you I hope you had it fun and uh it's already like very long so I apologize for that but I hope it was useful and yeah I will see you later