Usability Evaluation of the Local Large Language Models

Interaction Design
Department of Computing
Master's thesis
University of Turku

Author:
Teemu Kivimäki

11.6.2025
Turku

The originality of this thesis has been checked in accordance with the University of Turku quality assurance system using the Turnitin Originality Check service.

Master's thesis

Subject: Interaction Design, Department of Computing
Author: Teemu Kivimäki
Title: Usability Evaluation of the Local Large Language Models
Supervisor: Filip Ginter
Number of pages: 60 pages
Date: 11.6.2025

Local artificial intelligence (AI) features have recently gained remarkable attention, which indicates the growing need to recognize the existing challenges for wider adaptation of these technologies. Holistic approach is needed since the usability evaluation of locally installed LLMs requires a comprehensive view due to the challenges faced by different target groups and varying levels of configuration and implementation methods. Factors such as limited hardware resources and different user expectations need to be considered accordingly.

This thesis aims to recognize current advantages and challenges of using local large language models (LLMs) in average use environments such as homes and small to medium-size businesses and draw a broader view of the trend with methods such as heuristic evaluation of the user interfaces and by conducting a user study utilizing a general-use locally installed LLM (Meta Llama3-8B).

Regulations such as European Union’s General Data Privacy Regulation (Regulation (EU) 2016/679) create responsibilities for businesses to handle data in a secure and responsible manner. Likewise, the home users can also benefit from more confidential AI environments depending on their use cases. In this context, some of the key focuses of local LLMs strongly relate to privacy and data security. While using AI features, it is important that the end-user stays in control of their data and proper distinctions are drawn between local and cloud processing, as this approach allows users to make more informed decisions about their own data, thus enhancing a more user-centric approach to LLMs and AI more broadly.

Key words: Large language models, Local usage, Usability evaluation, Artificial intelligence

Table of contents

1 Introduction

Local artificial intelligence (AI) features as part of the consumer market are a relatively new phenomenon and have become more relevant as progress has been made in computer processing power. In the context of hardware accelerators, new neural processing units (NPUs) are specifically made for matrix operations and can work as effective alternatives for graphics processing units (GPUs) or central processing units (CPUs). Because of the rapid development of NPUs, big introductions of AI-powered computers have been recently made to the market, such as Microsoft's Copilot+ PCs[1], being marketed as the new AI-era for home computers. Although currently limited to certain exclusive computer series, predictions have also been made that eventually NPUs will be more standardized alongside GPUs and CPUs. This combined with the observation that GPUs and CPUs are improving in their performance over time as well, the notable progress in consumer hardware capabilities will enable large language models (LLMs) to be more topical and efficient for local usage.

The general usage of LLMs has grown rapidly in the past few years and the scope of different use cases are broad, ranging from very general questions to more personal matters. For example, it has been assessed that people are using OpenAI's GPT-4 and other LLMs for mental health support in the times of symptoms and hardship (Ganesan et al., 2024)[2]. Not only do such use cases incentivize to evaluate the accuracy of LLMs, but they also raise questions about the privacy and user control of the services. Locally installed LLM solutions become a relevant talking point in these discussions.

Despite the potential benefits of running LLMs on a local computer setting, factors such as accuracy, user experience and accessibility over existing online counterparts remain an open question in the field of interaction design. This thesis work examines the locally installed open LLMs from the standpoint of accessibility, responses provided to given questions (accuracy rate) and the user interface and user experience (UI/UX) for usage of the models. In general, analysing usability and accessibility factors of such models can be complicated and require a multi-perspective approach. Focus is drawn to developers, industries, students and regular users and the kind of challenges that different user groups may face to meet their set goals. The goal

[1] Microsoft (2025). Shop Copilot+ PCs. Windows. https://www.microsoft.com/en-us/windows/copilot-plus-pcs
[2] Ganesan, A. V., Varadarajan, V., Lal, Y. K., Eijsbroek, V. C., Kjell, K., Kjell, O. N., ... & Flek, L. (2024). Explaining GPT-4's Schema of Depression Using Machine Behavior Analysis. arXiv preprint arXiv:2411.13800.

is to discover suggestions to make local LLMs more accessible to the relevant user groups. More specifically, three research questions are formed for the thesis:

1. Which factors hinder the use of local LLMs for the relevant user groups?
2. How to make local LLMs more accessible to the relevant user groups?
3. Can local LLMs offer *significant* value to the relevant user groups?

First and second research questions are closely related to each other, although distinction still needs to be made between them, as it is possible that the discovered general improvements for accessibility (question 2) do not address all the discovered factors that hinder the usage (question 1). Third research question is more separate, since accessible local LLM (the objective for the question 2) which is easy to use may still not offer the kind of significant value that the user is seeking. This would be the case if there is easy access, but the model would not answer the questions presented by the user properly enough. It is also more subjective and varies from person to person what is considered significant value. In other words, questions 1 and 2 are meant to cover broader accessibility and question 3 is more about the general usability of the models in the context of the user's end-goals.

The existing evaluation methods for LLMs are examined and an experimental user study is conducted (n = 6) in which regular computer users are guided to use a locally installed, offline and open LLM (Meta Llama3-8B) for their personal assistance. The 8-billion parameter model is chosen for the test as it performs efficiently on the higher-end consumer PCs of today. The test group includes people from different backgrounds and fields, and they are requested to ask four (4) questions of their own choosing from the model in English, after which they report their satisfaction or lack thereof for the answers provided. The participants are allowed to ask one (1) optional follow-up question for each of these questions for clarification. The participants will then evaluate the performance of the Meta Llama3-8B model in this regard. Evaluation process puts focus on the accuracy of the responses provided to discover potential limitations in knowledge base for potential real-life use cases. For businesses, researchers and developers, it is important to analyse the ease or difficulty of integrating local LLMs into their operations and software. Factors such as detailed documentation and high customization options are important to achieve these objectives.

Since the benefits of running local LLMs are easily recognizable, the existing challenges are important to acknowledge. Because these models are running locally, communication with a knowledge base does not require an internet connection to occur, apart from downloading the models in the beginning and possibly updating them in the future with new information and capabilities. Depending on the specifics of the task, the user may benefit from smaller, locally installed models in comparison to using heavier online models if the smaller and locally installed model is equally capable of completing the said task. Offline use also means that the problem of limited server bandwidth is eliminated, and the amount of people using the model will not affect the overall performance – when the end-user's hardware is powerful enough and in good condition, the model will operate properly and is ready to be used.

It is worth mentioning that because the information is not sent over the internet, the risk of data breaches is reduced, and the privacy of the user is enhanced. This can be highlighted with the fact that many online services for LLMs would, in a regular use setting, save the chat histories of the user for convenience to the remote databases and link this history to the user. This considered, the user may feel more comfortable asking general sensitive questions such as medical, religious or political ones from the locally installed LLM, knowing that all the information is processed and stored on the device that they always have physical access to.

When AI is brought up more broadly in public discourse, the issue of trust is often raised as a concern. Such discussions may include lack of control over the use of AI. Local LLMs effectively increase user control and may even maximize it in this regard, which can serve as a good starting point for building more trust between the end-users and AI. Thus, such concerns can be properly addressed by offering relevant solutions to the users. Benefits also include the customization of LLMs according to user’s personal needs and making AI services more accessible in remote areas or circumstances where internet connectivity may be limited or unavailable.

1.1 Democratization of AI industry

The trust issues towards AI have led to the industry taking actions to “democratize” itself. There is not one definite answer for what democratization of AI means. Rather, the term refers to a series of various actions that are aimed to lower the threshold for the users to trust AI services. Democratization can be defined either based on the end-goals or specific taken actions. End-goals are perceived to lead to the user feeling that they are in control of the AI that they are using, rather than some outside entity. Taken actions to reach such an end-goal include, for instance, *open* models (transparency) and enabling local use of the AI-models.

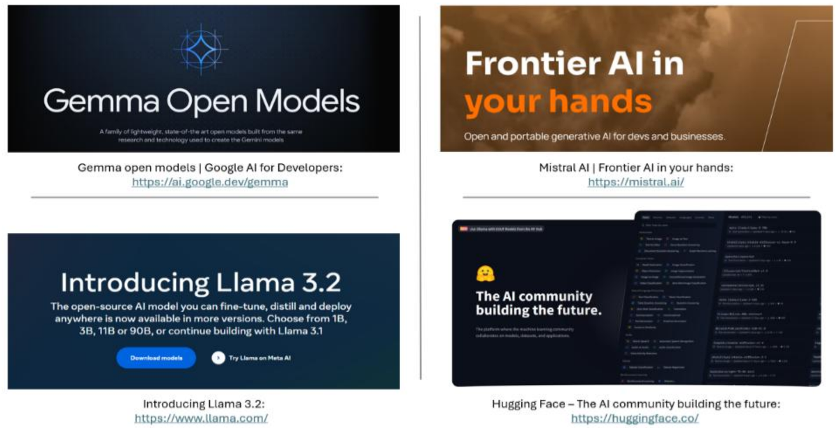


Figure 1. Democratization effects can be witnessed in marketing rhetoric for AI. Examples include Google's Gemma Open Models[3], Mistral AI[4] highlighted to be *in the hands of the user* and Meta's Llama[5] marketed as open-source model. Hugging Face[6] has been a strong force in AI community engagement.

AI democratization steps serve significant value to the public, as they address many legitimate trust concerns and issues. However, it is still realistic not to view democratization from the standpoint of common clichés, such as treating AI as some common good created for the people, even if such philosophical approaches would sound ideal. Fact is that the main players of the democratized AI field, namely big tech, are still seeking to profit from the created products and services and these steps are largely driven by commercial interests. With the reduction of the general trust problems towards AI, the creators of AI solutions can attract more users (customers) and ultimately create more revenue. While acknowledging this fact, it is worthwhile to note that this may not necessarily be something to be seen as a problem, as this can be a good scenario for all involved parties; the users get their valid trust issues addressed, and the providers gain more interest from this increased trust.

1.1.1 User empowerment and user-centered approach

User empowerment is an important concept in the context of AI democratization. It refers to circumstances in which users are making autonomous decisions and having a broader impact on their own user experience. Autonomous decisions mean that there will be as little guidance as possible from the service provider and the user is leading the interaction. In other words, user

---

[3] Team, G., Mesnard, T., Hardin, C., Dadashi, R., Bhupatiraju, S., Pathak, S., ... & Kenealy, K. (2024). Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.
[4] Mistral AI (2024). Frontier AI in your hands. Mistral AI. https://mistral.ai/
[5] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M. A., Lacroix, T., ... & Lample, G. (2023). Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.
[6] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., ... & Brew, J. (2020). Huggingface's transformers: State-of-the-art natural language processing. arXiv 2019. *arXiv preprint arXiv:1910.03771*, *10*.

empowerment enables user's potential to as full extent as possible. This can be achieved by providing broader user control and access to new approaches and opportunities.

When user empowerment is examined in the context of LLMs and other AI features, there are several important aspects to consider. For instance, LLMs and other AI tools can assist with real-life decision-making and empower users this way, which may in return create new opportunities for the user or make them think more broadly about problems and challenges they may encounter. Admittedly, online LLMs are also capable of this and therefore more relevant distinction between online and local implementations tends to rely on factors such as control and customization, which are enhanced in local AI solutions. Since the user is building the AI solution themselves by, for example, deciding which client they are going to use the local AI in and choosing from a large selection of models that are either general or tailored for their use cases, they can ultimately have this AI "environment" configured how they want it. This contrasts with online services where the environment, with possibilities and restrictions, is oftentimes pre-configured for the users.

Due to high amount of local LLMs out there for different use cases and the high customizability options based on user's specific needs, this can be considered a relatively user-centered approach in AI industry, which also drives user empowerment more broadly. However, proper analysis of the interaction design for local LLMs and more specifically user-centered design, will often require holistic thinking. This means that it is not enough to just analyse the selection of various models and their overall capabilities. Different existing frameworks and clients that enable the use of the models will likewise require examination, as these can also be either liberating or restrictive in different ways. For instance, local LLMs themselves do not provide any instructions for how the computer it is used on should run them. This is a task for the framework or client to handle. Therefore, if these are poorly optimized, there may be further restrictions for what models could be run, even if the computer does fulfil model's hardware requirements theoretically. To properly enable user empowerment with user-centered design, all steps for running local AI, including these technical specifics, need to be examined and implemented efficiently.

1.1.2 Timeline of AI to big tech

To understand the wider context of AI democratization, it is crucial to comprehend the historical context of AI. While the "birth year" of AI itself is difficult to pinpoint, several important steps can be recognized beginning from the last century. The history of artificial neural networks can be traced back to the 1940s, when McCulloch and Pitts created an algorithmic, computational model for neural networks (McCulloch, W. S., & Pitts, W., 1943)[7]. Turing Test by Alan Turing in 1950 was a major step in shaping the philosophical approach to AI and presented a wider question about the computer's ability to think (Turing, A. M., 1951)[8]. The Dartmouth Summer Research Project on Artificial Intelligence in 1956 is often referred to as the birthplace of AI as a research field (McCarthy et al., 1955)[9]. Such examples are just few of many but demonstrate the growing interest for AI that occurred in the 1940s and 1950s. However, due to the limited hardware capabilities at the time, this interest saw a decline, which continued for decades. What was theorized to be possible was quickly deemed impossible once these limitations were encountered. The interest increased gradually again when it was observed that AI capabilities are possible to implement not just in theory but also in practice.

A significant accomplishment for the deep learning capabilities was demonstrated in ImageNet Challenge competition in 2012 by AlexNet, which was an architecture of artificial neural networks for image recognition. During the competition, AlexNet accomplished a low error rate of 15.3%, which was a remarkable result at that time. In comparison, the error rate for 2nd place was 26.2%. (Krizhevsky et al., 2017)[10] AlexNet's good performance was due to Graphics Processing Unit (GPU) utilization during the training phase. As an outcome of this achievement, the potential of GPUs resulted in rising interest in deep learning. Likewise, the significant demonstrations in the early 2010s, such as AlexNet, were a big motivator for big tech companies to "hoard" the emerging AI technologies for themselves. More specifically, these companies saw AI as something to invest in that had a very reasonable chance of succeeding, based on these already proven capabilities. The circumstances were ideal for a situation to occur in which industry-leading AI solutions began to get more centralized around the big tech companies.

This centralization of the so-called "state-of-the-art" (SOTA) AI has caused noticeable trust issues, especially when one significant business model for these big tech companies involves making money from the user data. While these companies are not necessarily selling the user

---

[7] McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics, 5(4), 115-133.
[8] Turing, A. M. (1951). Can digital computers think?. The Turing Test: Verbal Behavior as the Hallmark of Intelligence, 111-116.
[9] McCarthy, J., Minsky, M. L., Rochester, N., & Shannon, C. E. (2006). A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955. AI magazine, 27(4), 12-12
[10] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. Communications of the ACM, 60(6), 84-90.

data to third parties, which may be prohibited by the existing data protection laws in the European Union (EU) region and elsewhere, they let third parties advertise on their platforms and target these advertisements based on the data these big tech companies possess. Personalized advertising, which is generally very profitable, requires that big tech companies are aware of users' interests in masses. Interests can be recognized by monitoring user activities on the internet, both on the websites that are owned by these companies and via tracking cookies embedded on the websites of the third parties. Tracking, which is necessary to enable personalized advertising, creates a complicated relationship between the users and the companies. Many users see privacy as a fundamental right and it is likewise important to note that consensus on privacy rights is established in multiple international treaties and documents, including under article 12 of Universal Declaration of Human Rights (UDHR)[11].

The existing business models, which include tracking user activities, cause users to be concerned about the general idea that the AI solutions and tools are in the hands of the big tech. The concern, more broadly, is that the same tracking practices would continue in a new environment where AI services are present and actively used, as this has been done by the same companies in the traditional environment for a long time. Furthermore, the common "assistant" nature of many AI features and the manners by which they are humanized to a certain extent, may cause people to use these tools and features in ways that they would not use the normal internet, such as search engines. One common example is how the models are sometimes used for personal therapy service with users explaining personal problems in life and seeking help. It is noteworthy that this rather significant change in overall user behaviour when interacting with AI in comparison to the rest of the internet could result in more personal questions and sensitive data being collected, as users may approach AI as if they are discussing with a personal assistant, therefore having more confidence in it. However, this confidence is based on trust, as the user cannot be certain how the data is accessed and used once it is sent over. The approach will always be based on trusting the provider of the AI service.

To tackle these challenges, AI providers can make further efforts to be more transparent to the public about the operations and handling of data. Alternatively, they can enable environments in which the users can be certain that outside access of any kind does not occur, such as allowing fully local usage of the AI tools and features.

[11] United Nations General Assembly. (1948). The universal declaration of human rights (UDHR).

2 Local LLMs

The usage of large language models (LLMs) can be divided into two categories: online and local. Online LLM refers to model access where inputs and outputs are processed on a remote server. Local LLM, in contrast, refers to access where inputs and outputs are processed locally on the user's own device. In general, there are not technical differences between the models that are used locally or online. The distinction only refers to how the models are being accessed. However, it is important to note that even though local and offline usage are often viewed in the same context and local LLMs would typically be used offline, a model which processes inputs and outputs locally may still not be fully offline. For example, if the user has an internet connection, technical limitations to access the internet for web searching to supplement the answer to the user's question do not exist, even if the model operates locally. In theory, this could still be considered local usage if all the input and output processing occur on the user's own device. Therefore, the distinction would require a four-level categorization.

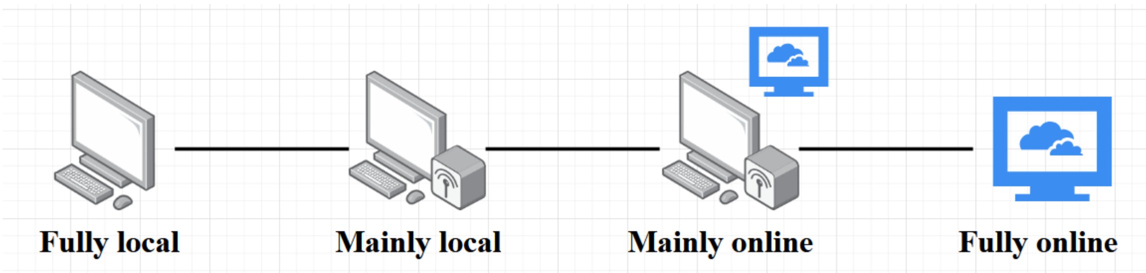


Figure 2. Four-level categorization of local and online AIs (visualized with draw.io[12])

Fully local AI can be used without any internet connectivity. This would include most local LLM usage and common frameworks. *Mainly local* AI includes usage in which inputs and outputs are processed locally but some kind of internet connectivity is still occurring on the background. For instance, Microsoft's Image Creator for Copilot+ PCs is an AI service in which inputs and outputs processed are locally on the Neural Processing Unit (NPU), but the program is still connected with Microsoft account and uses cloud services to ensure that the local AI usage complies with Microsoft's content policies and responsible AI guidelines. *Mainly online* AI is the kind of usage that takes noticeable advantage of the user's computer resources for the operation, but the main processing still occurs on the cloud. *Fully online* AI would rely entirely on the cloud for the operation, e.g. ChatGPT. Unless specifically stated otherwise, local LLMs in this paper refer to *fully local* LLMs.

[12] JGraph. (2025). diagrams.net, draw.io. https://www.diagrams.net/

LLM clients may allow files for inputs. File extensions provide instructions for operating systems to launch the correct program that can interpret the contents of the file. Technically all the files are in text form but can only be read by specific programs and file extension is merely an instruction for the operating system to open a program that can read this content. While Microsoft Word can read the text contents of a .doc file, similar capabilities can be likewise integrated into the LLM clients. The client will transform the file into text, which the text-based model can then utilize. This allows prompts such as ones where the end-user requests the LLM to enhance or fix something in their Word document and the LLM can the provide the modified version as the output. This flexibility enables broad ranges of use cases, which can be examined both in the context of online and local LLMs.

2.1.1 Licensing options

A common misconception is that if something is free and open, it can be used freely without any limitations. However, this is not the case. Legally, it is important to note that the users don't "own" any of the locally installed LLMs on their computers and they are merely licensed to the end-user (legally referred to as 'licensee') when downloaded and used. The same approach applies to any software and digital tools that are intellectual properties (IPs) and distributed to the users. This contrasts with public domain (PD) releases, which are not considered IPs under the legal framework and therefore can be freely used without separate licenses. In practice, licensing means that there may be limitations on, for example, the commercial use of the models and other similar restrictions related to the usage. Some models, such as Llama, may allow commercial usage to a certain point, after which a separate license will be required. In Llama's case, this limitation is set at 700 million active monthly users for any service, after which the model cannot be used, modified or redistributed under the standard license (Kshetri, 2025)[13].

Model providers, legally referred to as 'licensors', want to enable broad commercial usage with the standard license as it drives interest in the models and innovation, which is also in their own interests. However, once the businesses grow big enough, a separate license may have to be agreed on. Furthermore, the price tag of this license might not be known in advance but rather negotiated with the parties on a case-by-case basis. The licensor, which is the model provider in this case, may be able to unilaterally change parts of the license agreement if the agreement itself makes such changes possible (this is known as modification clause). Likewise, the

[13] Kshetri, N. (2025). Economics of Open Source Software and AI Models. *Computer*, *58*(1), 134-141.

licensor may be able to terminate the license based on either given list of reasons provided to the licensee or for any reason if the agreement itself allows such actions. The agreement should outline all these details. Depending on what the end-user is planning to do with the model, which requires approval in a form of a license, being aware of different licensing options is important, as they can vary between local LLMs from different providers.

2.1.2 Different types of models

Models can be divided into multiple categories. For example, they can be instruct- or chat-based, natively (independently) trained or distilled from another model, and they can also be created for general purpose uses or tailored for specific use cases.

Common model categories		
Instruct-based (single-turn)	**OR**	Chat (multi-turn)
Natively trained (independent)	**OR**	Distilled (from another model)
General purpose	**OR**	Tailored

Table 1. Distinctions between large language models.

The categories are, in every instance, fundamentally linked to how the models are trained. For example, chat-based (multi-turn) models are trained with dialogue transcripts, whereas instruct-based (single-turn) models are trained on instruction-response basis. Whether the model is general purpose or tailored depends specifically on the category (field) of the training data and whether the model is natively trained or distilled is related to the training approach as well. Each category is relevant, and one category is not objectively better than the other. The choice between the categories depends on what the end-user is trying to accomplish and what the personal preferences are. For instance, general purpose model may be able to answer a question of a specific area of knowledge in a way that is sufficient for the user. However, once the user gets into even deeper specifics of the field, eventually tailored model may become more preferrable. In the context of general home usage, categories can also be seen as trade-offs when various hardware limitations are being considered – as an example, instruct-based models may achieve smaller size when conversational knowledge is not needed in comparison to the chat-based models. Models of smaller size would then be lighter to run on a consumer hardware. For such reasons, understanding various model categories is also important for comprehending the overall usability and capabilities of locally installed models on limited hardware configurations.

2.2 Advantages

The most significant advantages of running local large language models (LLMs) compared to the online counterparts relate to data security, privacy and increased transparency. Local LLMs can be used to handle sensitive datasets in environments in which such advantages are either mandatory (required) or wanted (optional).

2.2.1 Privacy and increased openness

By default, local LLMs are private from outside access due to the nature of their operation. Both the input processing and output generation occur locally, and this factor creates new potential and approaches to AI usage. The advantage can be quite significant for institutions and businesses, as they may have an interest in using AI features such as LLMs to process information which may be subject to data security and privacy laws, such as EU Regulation 2016/679, commonly known as EU's General Data Protection Regulation (GDPR). The 6th principle of GDPR mandates appropriate security measures and protection against unauthorized or unlawful processing of data (2016/679)[14]. GDPR, together with other laws and regulations regarding data security and privacy, result in a circumstance in which using online (e.g. cloud-based) services for handling and processing protected data is often not ideal for institutions and businesses. In many cases, this processing may be prohibited altogether. Once the data leaves the "house", the operator becomes uncertain who might access it in the future, while still carrying legal responsibilities over that data. Such issues are addressed once LLMs are used in a manner where all the protected data stays in-house, remains secured from unauthorized access and therefore complies with the existing laws and regulations, such as GDPR. For these purposes, local LLMs are an ideal solution.

It is important to note that the client which runs the local LLM may still introduce various online features to improve the user experience. Some clients may include both local and cloud models. Therefore, for any entity or person that is legally required to protect the data, it is still relevant to make sure that such additional features are not in use or included. In any case, keeping the device disconnected from the internet is still typically advisable in cases where full data protection is either required or desired. In addition, local implementations place a lot of responsibility on the end-user. More specifically, the subject of privacy relates to the wider

[14] Regulation, P. (2016). Regulation (EU) 2016/679 of the European Parliament and of the Council. Regulation (eu), 679, 2016.

question of cybersecurity, as there are many operators that would actively threaten user privacy and security, even in cases where all the data is stored locally. For instance, researchers at Hudson Rock discovered that in the context of cybercrime, infostealer infections as form of malware have increased by 6000% from 2018 to 2023, which indicates a major shift in the common attack vectors[15]. These findings demonstrate that local systems need to be protected, just like cloud solutions. Infostealers are particularly challenging because oftentimes the user would not know that the system is infected. Infostealer malware would, in practice, seek sensitive information from the hard drive and send it over to the threat actors. Nowadays such information would often be found from the browsers, e.g. login credentials, cookies, stored sessions, search and browser histories. However, once local LLMs become more common and are being used to handle sensitive sets of data, it is not out of the question that infostealer variants will be created to target this stored data as well. Therefore, the users of the local AI still need to exercise reasonable security practices on their devices to make their usage truly secure and private. This considered, local AI software on a compromised home or enterprise system that is connected to the internet may, in the bigger picture, be perceived less secure and private than using trusted online counterparts with a system that is not compromised.

Open-source approaches can offer notable advantages for deploying LLMs, such as in context of transparency, reproducibility and ethics (Manchanda et al. 2024)[16]. However, when "openness" of the models and AI systems are examined more broadly, there often tends to be lack of precision. (Widder et. al, 2024)[17] As one example, it is important to separate open-weight models from open-source approaches. In the context of open-weights, pre-trained weights of the model are publicly available, while the training data may not be.

Open-weight models are often perceived to be open-source models. Technically speaking, this is only the case if all the details related to the model, including training data, are publicly accessible. Most mainstream, industry-leading open LLMs are open-weights but not open-source. There are rather practical reasons for this such as commercial interests, intellectual property (IP) protections juridical issues. The latter one considered, even if a privately owned AI company wanted to be transparent and release these details to the public, the data the model

[15] Hudson Rock. (2023). 100,000 Hackers Exposed from Top Cybercrime Forums. Hudson Rock. https://www.hudsonrock.com/blog/100-000-hackers-exposed-from-top-cybercrime-forums
[16] Manchanda, J., Boettcher, L., Westphalen, M., & Jasser, J. (2024). The Open Source Advantage in Large Language Models (LLMs). arXiv preprint arXiv:2412.12004.
[17] Widder, D. G., Whittaker, M., & West, S. M. (2024). Why 'open'AI systems are actually closed, and why this matters. Nature, 635(8040), 827-833.

was trained with may be protected with copyrights held by third parties and therefore cannot be shared legally. Additionally, since some models may be trained with trillions of tokens of data, in many cases sharing the training data would be either impractical or impossible. In this context, it is important to note that open models (with open-weights approach) do not address the existing black box issue of AI because there is lack of public knowledge as to how the models were trained, which contributes to a situation in which full transparency cannot be achieved. Black box refers to the phenomenon where users of AI cannot perceive what the output is based on, in the context of LLMs and other AI software. For the general userbase, these factors are somewhat insignificant, as it is reasonable to think that most home users are not interested in the technical details of the training data. However, once bigger and more impactful decisions are made with the assistance of AI and accountability to third parties becomes a factor, it could end up being more important to have further transparency in terms of overall data selection for model training. In addition, the transparency requirement of GDPR can present further challenges for full adaptation of AI software, including local LLMs.

Even though models can be considered "open", whether openness refers to open-weights or open-source, the environment in which the open model is used in can still be closed-source. More broadly this relates to the whole operating system environment and less broadly to the LLM client. For instance, the most used desktop operating system, Windows, is not open-source (Awan & Khan, 2022)[18]. In the same manner, some popular local LLM clients may not be open-source. For users who are interested in such openness more generally, these factors may be important to consider. Both open-source operating systems (e.g. Linux distributions) and open-source LLM clients exist, so it is possible to make choices considering all relevant software in the environment. While it is noted that lack of precision is an encountered issue with defining and drawing consensus on the concept of openness more broadly, local LLMs still have a clear potential of increasing openness on a spectrum rather than being viewed as fully open (open-source) solutions, i.e. open-weights approach that many local LLMs follow can be seen as an improvement in openness in comparison to fully closed-source solutions while still being less "open" than fully open-source alternatives.

In addition to the enhanced privacy of using local LLMs, research is being conducted on privacy-preserving potentials of model training as well. Chen et al. (2023) introduced federated

[18] Awan, M. T., & Khan, K. (2022). Linux vs Windows: A Comparison of Two Widely Used Platforms. *Journal of Computer Science and Technology Studies*, *4*(1), 41-53.

learning (FL) as a technology that aims to decentralize training process of LLMs. FL is a collaborative model learning approach where clients share asynchronous updates with each other, such as gradients and weights, while the raw data is kept stored locally.[19] Adopting such approaches could avoid centralization of the training data and therefore safeguard the privacy of the involved parties that take part in the training procedure. Therefore, there are concrete ways to enhance the privacy of the model users and those who participate in the model training. Such practices could be mutually inclusive and aim to collectively address wider privacy issues regarding LLMs and other AI software.

2.2.2 Speed and performance

Even though local LLMs are considered faster than online counterparts with ideal hardware configurations, speed is not typically considered as significant of an advantage for common use cases as privacy or increased openness is, although worth covering. In cases where the user gets a good output from the LLM to their question, it is generally less relevant to them whether the response generation, for example, takes 10 or 20 seconds. What is more relevant overall is that the operation does not “challenge” user’s patience, i.e. it is simply fast enough. However, one considerable advantage of local LLMs is that they are not tied to server uptime or bandwidth in the same way online LLMs are. In contrast to local LLMs, speed and performance of any online LLM is ultimately determined by whether the remote server is under load or not and in extreme cases, the service may be completely inaccessible. With some services, users may be able to pay for priority access, which helps with accessing the service if only free users are being "throttled". Despite this possibility, priority access may still be restricted under heavy load and if the servers are completely down, no one can access the service, regardless of whether the users are under free or paid (priority) plans.

Depending on the specific use cases, faster speed of local LLMs may be a trade-off in comparison to the model quality and capabilities. This is because the models that are smaller in size are faster with output generations than heavier models on the identical hardware configurations. Ideally, users of limited hardware capabilities, e.g. home users, should aim to use the smallest model available that can successfully complete their tasks if they want to achieve the fastest possible output generation while still maintaining the required and expected quality levels. However, patience may result in better outputs when heavier models are used

[19] Chen, C., Feng, X., Li, Y., Lyu, L., Zhou, J., Zheng, X., & Yin, J. (2024). Integration of large language models and federated learning. Patterns, 5(12).

depending on the use case, which can be necessary for certain tasks. Because "performance" can refer to either the speed of the model or the quality of the model, it is important to understand the balance when considering and evaluating the overall model performance. Improving the performance on one side (speed) can negatively impact the performance on another (quality of the model) or vice versa improving the performance on quality can negatively impact the performance on speed of the output generation. However, it is important to consider that heavier models might not be needed to complete certain tasks, or the quality differences are generally so minimal that these could, in practice, be considered insignificant.

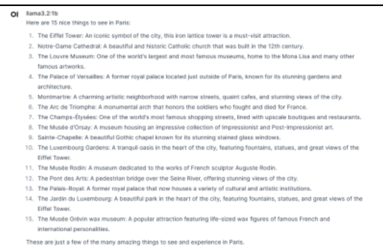
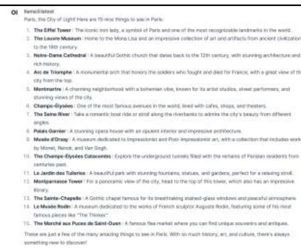
<table>
<tr><th colspan="3">Speed variance for local LLMs – "List 15 nice things to see in Paris"</th></tr>
<tr><td>Llama3.2-1B
(Q8_0)</td><td>Llama3.2-3B
(Q4_K_M)</td><td>Llama3-8B
(Q8_0)</td></tr>
<tr><td>
</td><td>
</td><td>
</td></tr>
<tr><td>response_token/s: 122.68 tokens
prompt_token/s: 14234.22 tokens
total_duration: 3417.67ms
load_duration: 13.48ms
prompt_eval_count: 62
prompt_eval_duration: 4.36ms
eval_count: 417
eval_duration: 3399.2ms
approximate_total: 3s</td><td>response_token/s: 89.97 tokens
prompt_token/s: 7706.67 tokens
total_duration: 5778.19ms
load_duration: 13.03ms
prompt_eval_count: 52
prompt_eval_duration: 6.75ms
eval_count: 518
eval_duration: 5757.78ms
approximate_total: 5s</td><td>response_token/s: 48.02 tokens
prompt_token/s: 3111.42 tokens
total_duration: 10776.68ms
load_duration: 12.47ms
prompt_eval_count: 58
prompt_eval_duration: 18.64ms
eval_count: 516
eval_duration: 10745.06ms
approximate_total: 10s</td></tr>
</table>

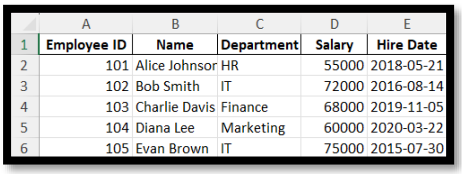
Table 2. Outputs generated on the same hardware configuration. Generation with Open WebUI[20].

In the speed test demonstration outlined in table 2, all three compared Llama models are capable of listing 15 things to see in Paris, which means that quality differences do not occur between models for this specific task or alternatively the quality differences are so insignificant that the end-user would likely not notice them when using the models for the task. Therefore, from the end-user's perspective, running the heavier model can be less ideal for this task since the 1B model is approximately three times faster and completes the task as successfully as the (slower) 8B model does, i.e. the larger model could be considered "overkill" for this task. However, it is worth noting that the difference in speed between these three models is somewhat minimal, and therefore debatable whether the margin of *few* seconds would challenge user's patience. In the big picture, the output generation in all three test cases is still relatively quick.

[20] Baek, T. (2025). open-webui: User-friendly AI Interface. GitHub. https://github.com/open-webui/open-webui

2.2.3 Data handling capabilities

The usability of local LLMs should not be solely evaluated based on the knowledge base. Even if the model may have shortcomings in its knowledge, data handling capabilities are important to consider. Knowledge limitations are less relevant in this context since the user is providing most or all the information in the input. With such tasks, the requirements from the model are more related to its abilities to follow instructions and print outputs based on the data.



	A	B	C	D	E
1	Employee ID	Name	Department	Salary	Hire Date
2	101	Alice Johnsor	HR	55000	2018-05-21
3	102	Bob Smith	IT	72000	2016-08-14
4	103	Charlie Davis	Finance	68000	2019-11-05
5	104	Diana Lee	Marketing	60000	2020-03-22
6	105	Evan Brown	IT	75000	2015-07-30

To demonstrate this, an AI-generated Excel worksheet (by ChatGPT[21]) was created. The sheet is a simple file which contains imaginary employees and their salaries.

The task for the model is to sort the employees by salary in descending order (highest to lowest). The modified data should then be printed as an output. For the task, text-only Llama 3.2-3B Chat 8K is used with AnythingLLM[22] client. The input and output are processed on neural processing unit (Snapdragon ® X Elite – X1E80100 Qualcomm® Hexagon™).

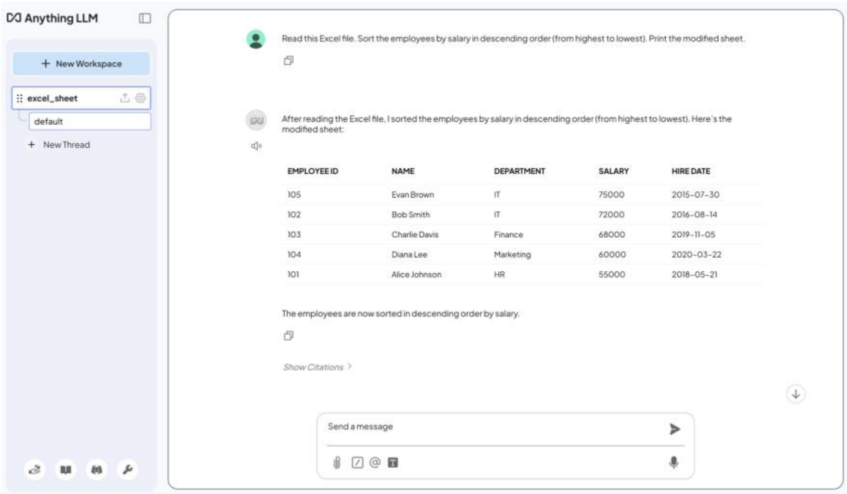


Figure 3. Llama 3.2-3B Chat 8K completed the task successfully.

Different clients may handle file uploads differently. Text-based (text-only) models are not capable of providing files as outputs but text-based table outputs can typically be copied into Excel with small adjustments. In this demonstration, hypothetically sensitive personal information (employee details) is being handled and sorted locally. Local LLMs, including smaller variants (3B in this demonstration) can be good for these use cases.

---

[21] OpenAI. (2025). ChatGPT (May 2024 version) [Large language model]. https://chat.openai.com
[22] Mintplex Labs Inc. (2025). anything-llm. GitHub. https://github.com/Mintplex-Labs/anything-llm

2.3 Common ways of running local LLMs

There are many ways of running local LLMs. Users should be aware of different options and consider their priorities and expectations regarding the overall user experience. For example, one factor to consider is whether the user wants a graphical user interface (GUI) in which to use the model or whether they prefer a more traditional command line interface (CLI).

2.3.1 Ollama, Command Prompt & Terminal (CLI approach)

Local LLMs can be run directly from command line interface (CLI). In this case, the user would have a framework installed for running local LLMs (such as Ollama[23]), which is then called with a command in the Command Prompt (CMD). Using the same command, Ollama framework can also be called with PowerShell on Windows 10 and 11 or with macOS and Linux Terminals, making it a widely available option for many operating systems. Although running LLMs via CLI may be beneficial performance-wise since graphical user interface is not deployed, this comes at a cost of user-friendliness and lack of visual appearance. The environment in question also restricts the use of LLMs as, for example, pictures or files cannot typically be uploaded in such environments.

The way Ollama works is quite straightforward: the application programming interface (API) uses the local port 11434 by default. The model is then loaded into the memory and any program on the device can communicate with the loaded model by utilizing this local port. This enables creation of different software that use Ollama, making it a highly customizable option for both consumer and business use cases. Due to high customizability options, frameworks such as Ollama are also being used in LLM server environments.

In general, Ollama is a relatively popular method for running local LLMs. Many model collections in the library have been pulled (downloaded) millions of times. As of February 2025, Llama 3.1 has been pulled over 24 million times[24] and DeepSeek-R1 has been pulled over 19 million times[25]. Higher pull numbers are generally achieved with model families that come with smaller sized versions as well (either *distilled* or *natively trained* models), which allows use among the home users and therefore "popularizes" these model families.

---

[23] Ollama. (2025). Ollama. GitHub. https://github.com/ollama/ollama
[24] llama 3.1 | ollama. https://ollama.com/library/llama3.1 February 2025.
[25] deepseek-r1 | ollama. https://ollama.com/library/deepseek-r1 February 2025.

To use Ollama from CLI, users need to download Ollama framework. Ollama is available on Windows, Linux and macOS. After the framework is downloaded and installed, it can be called from Terminal or CMD – for example, with commands to download (*pull*) and use (*run*) LLMs. Below are the steps for installing and running LLMs via Ollama CLI.

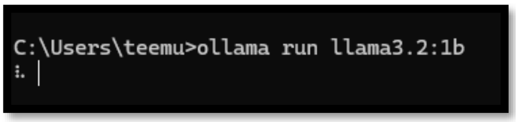
Ollama CLI - Model installation

ollama pull *model_name*

User is downloading (*pulling*) the 1b version of Llama 3.2 via Ollama, using Terminal.

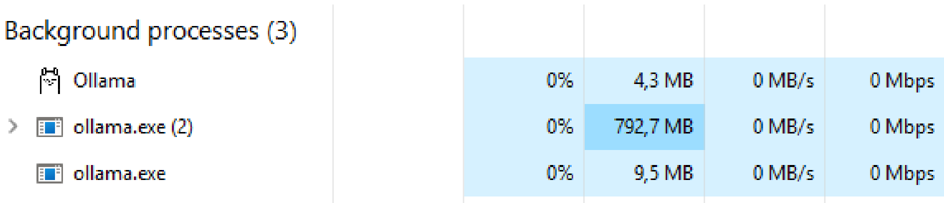
Ollama CLI - Loading installed model into the memory

ollama run *model_name*



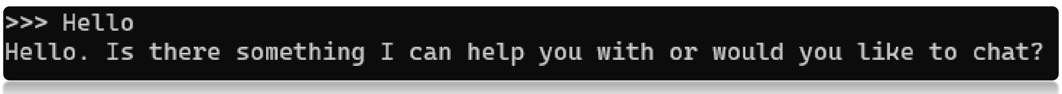
The screenshot on the left shows the loading status and image on the right shows the status when the model is loaded (into the memory). Loading time depends on model size and hardware speed.

Ollama CLI – Model loaded into the memory



Because Llama 3.2:1b (Q8_0) is a relatively small model in size (parameter count ~1,24 billion) it is moderate on memory requirements, in this case requiring less than 800MB of RAM and approximately 400MB of VRAM.

Ollama CLI – Interacting with the model



After the model is loaded into the memory, it can be used for text generation.

Table 3. Model installation, loading and interaction (with Ollama CLI).

2.3.2 LLM clients (heuristic evaluation)

Another relatively common way of running local LLMs is to deploy a personal-use graphical user interface (GUI) for usage, which is done with the LLM clients. While some clients prioritize the general userbase and ease-of-use, some clients are tailored towards more professional (e.g. programming) use cases. Depending on the client, the program may serve as a one-in-all package so that it contains everything the user needs to download and use the models. Some of the clients include their own model libraries. Clients are especially useful due to many additional and useful features they typically include in comparison to the previous CLI approach. For example, many clients allow users to upload files for LLMs to process. Three popular LLM clients will be covered in the section: GPT4All, Jan and LM Studio with closer look into the interaction design by using Jacob Nielsen's 10 Usability Heuristics for User Interface Design[26]. These heuristics are commonly adopted general and standardized principles in interaction design that serve as baselines for designing user interfaces.

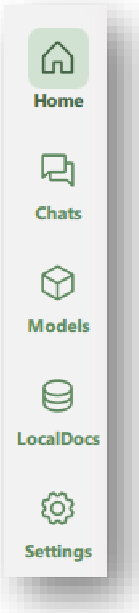
Jacob Nielsen's 10 Usability Heuristics for User Interface Design	
1. Visibility of System Status	• Communicate the status • Present feedback • No action with consequences without informing the user
2: Match Between the System and the Real World	• Speak user's language • Don't *assume* your understanding matches user's
3: User Control and Freedom	• Undoing, redoing • Cancel (exit current interaction)
4: Consistency and Standards	• Learnability through consistency • Established standards
5: Error Prevention	• Prevent users from making errors
6: Recognition Rather than Recall	• Reduce what users need to remember
7: Flexibility and Efficiency of Use	• Shortcuts, personalization, customization
8: Aesthetic and Minimalist Design	• Focus on the essentials, support goals • Avoid unnecessary elements
9: Help Users Recognize, Diagnose, and Recover from Errors	• Tell users what went wrong in clear language • Offer solutions
10: Help and Documentation	• Easy to find/search documentation

Table 4. Usability heuristics, including Jacob Nielsen's, form guidelines for user interface design.

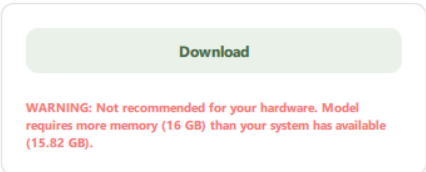
[26] Nielsen, J. (10, April). Usability heuristics for user interface design.

GPT4All (by Nomic)[27]

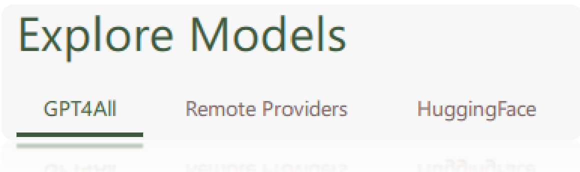
GPT4All by Nomic AI follows a very minimalistic design (8) approach for LLM clients. Despite this, it's still a relatively feature-rich client with several configuration options. The client is divided into five scenes: Home, Chats, Models, LocalDocs, Settings.



Home scene is the main view of the application upon launching it. It contains the latest news for the application, highlighting changes with the recent updates. The Home scene also includes quick links to Chats, LocalDocs and Models and the ability to subscribe to a newsletter. The bottom-panel includes links to documentation, Nomic AI social media pages and GitHub. **Chats** scene contains the saved chats with LLMs. **Models** scene lists all the installed models, with '+ Add Model' button taking the user to the GPT4All Model library, in which models can be installed with one-click approach. **LocalDocs** scene allows users to add collections of local documents (folder paths) which can be used for inputs. **Settings** scene includes Application settings, Model settings and LocalDocs settings.



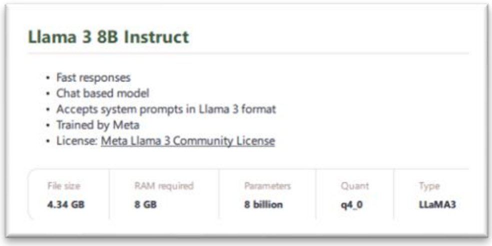
GPT4All puts a lot of emphasis on tailoring the user experience for home use. As an example, the main model library for GPT4All does not include models that are typically too heavy to run in these environments. Even with the lighter models, the model will warn the user if they are about to download a model that is not recommended for their hardware. This approach is consistent with Nielsen's Usability Heuristics, more specifically 1, 2, 3 and 5. Users are being informed about the consequences of the actions taken (1), the warning speaks user's language (2) and the user is potentially prevented from making an error, such as running a model that is too heavy for their hardware, resulting in poor user experience (5). However, despite this warning, the user is still able to download the model, therefore the warning is not a barrier for user control and freedom (3).



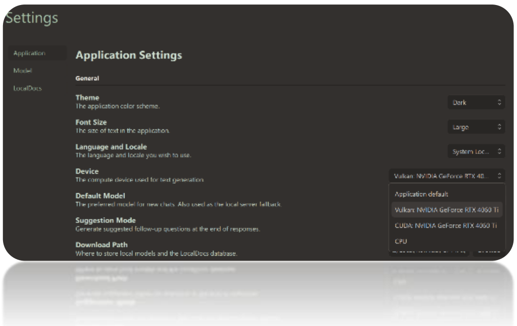
Users can directly download models from HuggingFace, including the larger ones not meant for home usage instead of the GPT4All-recommended models, which further improves

[27] Nomic, Inc. (2025). nomic-ai/gpt4all. GitHub. https://github.com/nomic-ai/gpt4all

User Control and Freedom (3) whenever needed. If the user wants to run cloud models with API key, this is also possible under ‘Remote Providers’ tab.

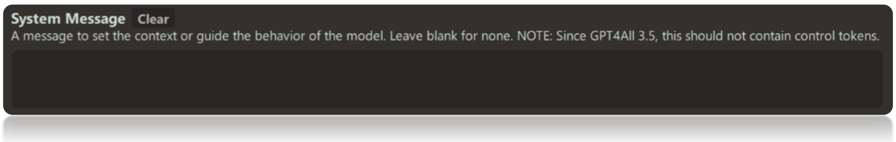


Each listed model includes essential information such as the model type, file size, parameter count, required RAM, quantization technique and the license under which the model can be used, highlighting Consistency and Standards (4).

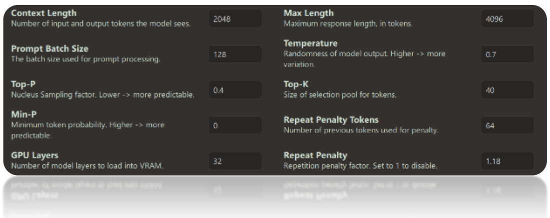


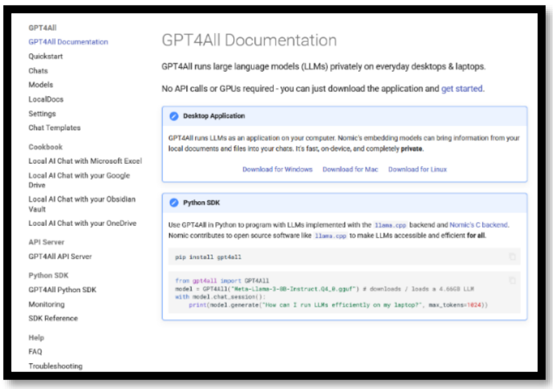
The application includes several common customization options such as Dark Mode, larger font size, changing language/locale or the device (e.g. Vulkan/CUDA GPU, CPU) that the user wants to use for text generation. These features enhance Flexibility and Efficiency of Use (7).

Furthermore, ‘Model’ section of the settings allows the user to create system messages for installed models, which sets the context and guides the behavior of the model.



The model settings are extensive and include a wide variety of configuration options, allowing the user to have significant impact on their own user experience, further improving User Control and Freedom (3) and Flexibility and Efficiency of Use (7).

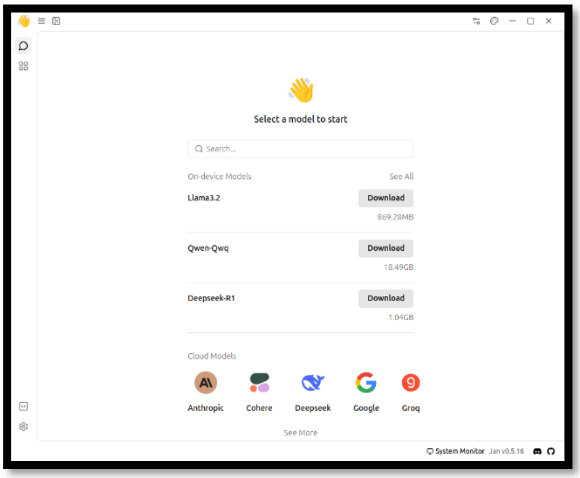




GPT4All includes documentation (10). Apart from documenting the application, the documentation includes integration options with commonly used environments such as Microsoft Excel, Google Drive, Obsidian Vault and OneDrive. API Server features and Python SDK are also documented.

GPT4All is available for Windows (x86 and arm64), macOS and Ubuntu.

Jan (by Jan.ai)[28]



Compared to GPT4All, Jan appears to follow an even more minimalistic design (8). For example, unlike with GPT4All, the menu selections are not accompanied by text. The main scene is clean, and the client draws clear distinctions between on-device models and cloud models. It is good to see the client listing a lightweight local model (Llama 3.2) as the first recommended option.

Whereas GPT4All has a taskbar on the left side of the client covering all the navigation within the client, Jan has a separate top navigation bar. Whichever approach is preferred is up to the end-user.

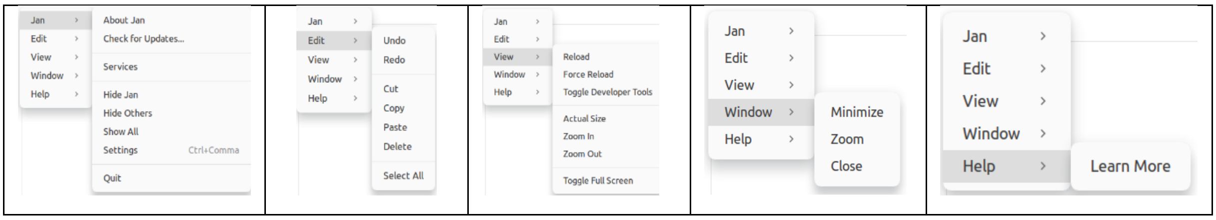
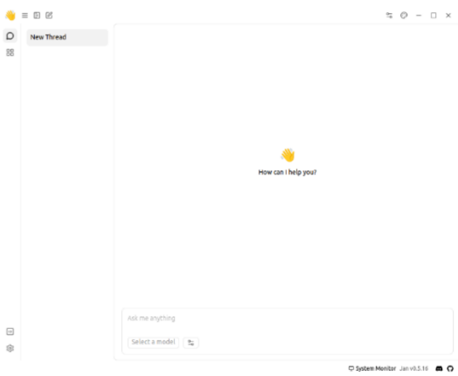


Table 5. Top navigation bar options for Jan (Jan, Edit, View, Window, Help)

The taskbar on the left side of the client includes icons to **Home** scene, **Hub** (model library) scene, **local API server** scene and **Settings** scene. Once the user has downloaded a model, the Home scene transforms from the Welcome screen (as shown in the image above) to the Thread scene.



From Thread scene, the user can then manage their chat history and select a model. The settings button ( ) next to model selection opens a taskbar to the right side which include assistant features (e.g. pre-prompt for the model), interface settings, model settings and engine settings. This can be a useful feature, as the user does not have to leave the chat interface to change these settings (7).

[28] Menlo Research. (2025). Jan. GitHub. https://github.com/menloresearch/jan

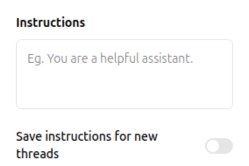
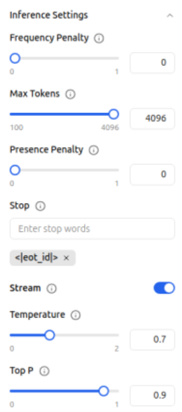
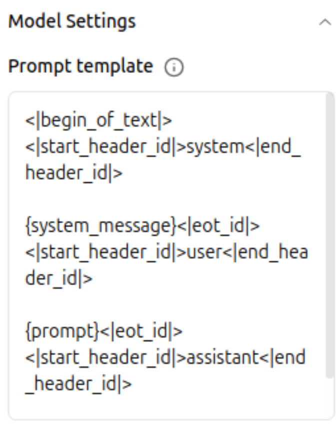
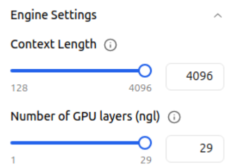
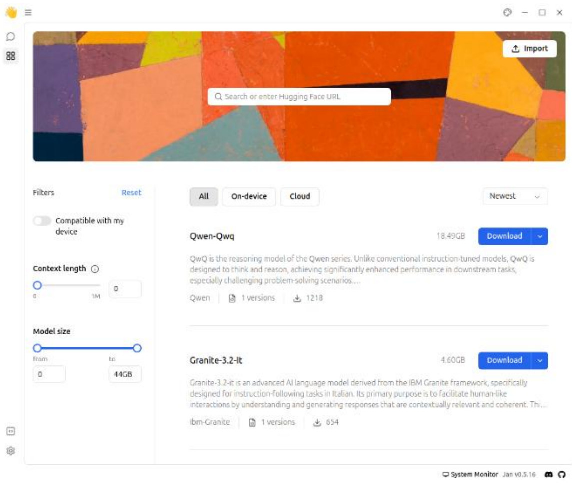
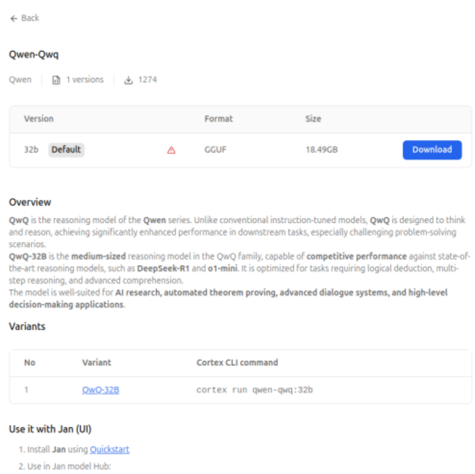
Jan's LLM configurations (chat interface settings)			
Assistant	Interface Settings	Model Settings	Engine Settings
			

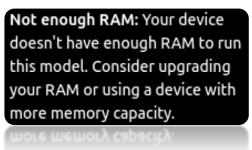
Table 6. Jan's chat interface settings.

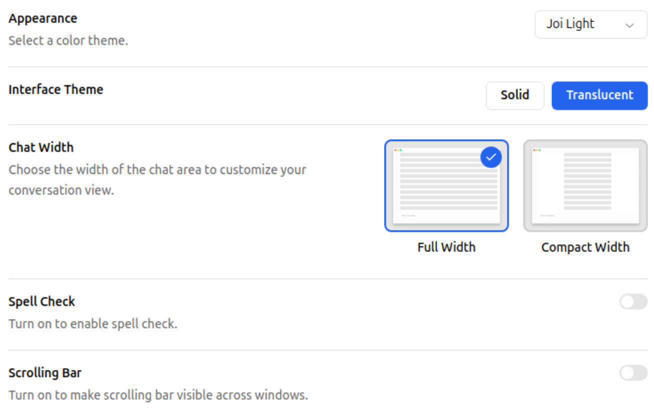


Unlike with GPT4All's model library, Jan does not issue warnings below the download button if the model is not recommended for the hardware. Instead, the library includes a toggle button 'Compatible with my device', which can be enabled separately (3, 5). This will filter out unrecommended models. The library lists newest models first, with the second option to sort by most downloaded.



Clicking on the model takes the user to a view where more details about the model are provided. Models are extensively covered (10). Compared to GPT4All, some information such as RAM requirement is not directly mentioned in numbers, but the model page will have a red warning sign (⚠) if the hardware does not meet the requirements (5).





For flexibility and ease of use (7), Jan offers multiple customization options in the settings interface under 'Preferences' tab. These include switching between interface theme (solid, translucent), changing chat width, enabling/disabling spell check or making scrolling bar visible. Jan includes four types of color themes: Dimmed, Dark, Light, Blue.

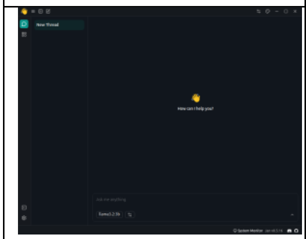
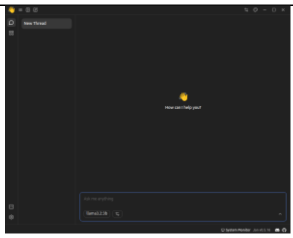
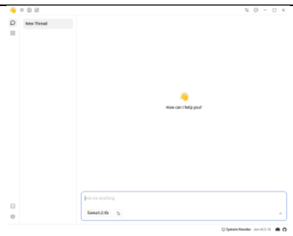
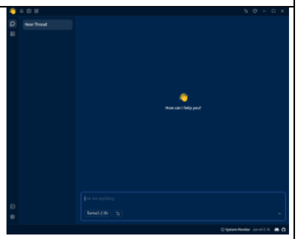
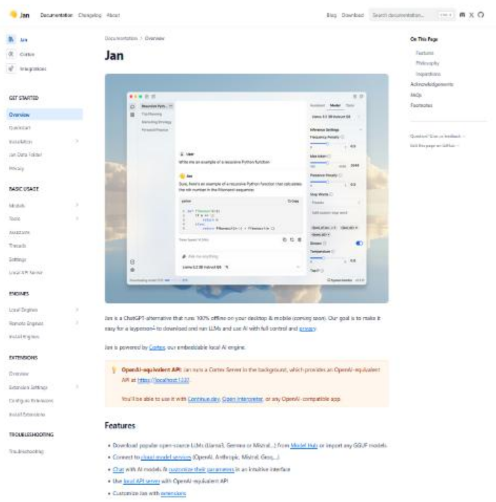
Color modes for Jan			
Dimmed	Dark	Light	Blue
			

Table 7. Four customization options for Jan client.



Jan includes documentation (10).

The documentation is divided into 5 sections: **Get Started** (Overview, Quickstart, Installation, Jan Data Folder, Privacy), **Basic Usage** (Models, Tools, Assistants, Threads, Settings, Local API server), **Engines** (Local Engines, Remote Engines, Install Engines), **Extensions** (Overview, Extension Settings, Configure Extensions, Install Extensions) and **Troubleshooting** (Troubleshooting).

Jan is available for Windows (x86), macOS and Linux. For Linux installations, the download is available as *AppImage* and *deb* packages.

LM Studio (by Element Labs Inc.)[29]

Upon first startup, LM Studio includes an onboarding scene, which provides a step-by-step guide to the user through their first action of downloading a model. The process is highlighted as "*downloading your first LLM*".

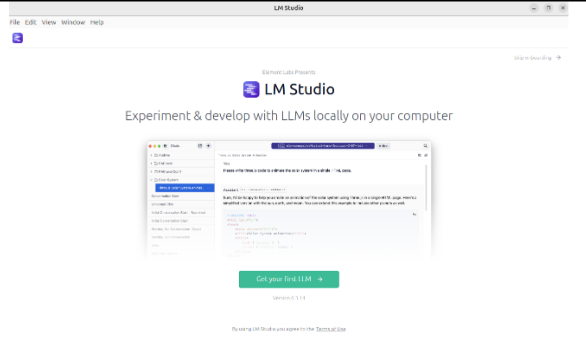
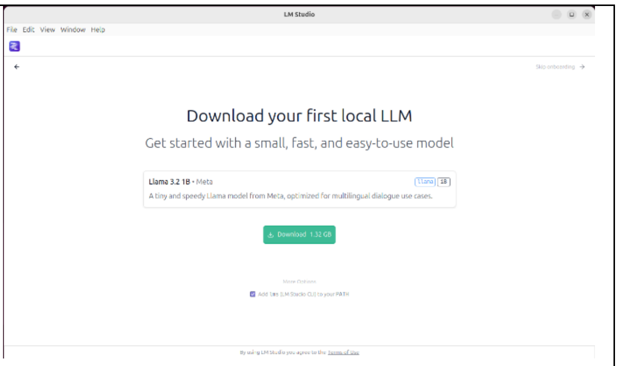
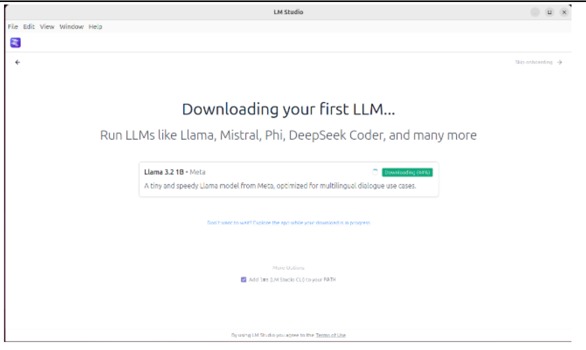
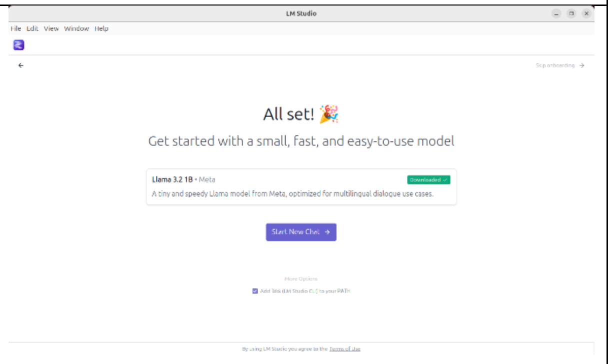
 (1). The user clicks "Get your first LLM" button.	 (2). The program recommends "*a small, fast, and easy-to-use model*" (Llama 3.2-1B in this case)
 (3). The user downloads the model.	 (4). The program informs the user that the download is complete and provides an option to start chatting.

Table 8. Four-step onboarding process during first launch of LM Studio.

The onboarding process demonstrated in table 8, which is essentially 4 steps (1. click → 2. recommendation → 3. download → 4. complete), is simple yet an efficient way to introduce the user to the process of downloading models. From the perspective of usability heuristics, this process guides the user and keeps them on track by communicating the status (1), the process speaks users' language (2) by being easy to understand and by referring to user's first downloaded LLM and it is consistent (4). In addition, the user is allowed to skip this onboarding process if they wish, thus onboarding scene does not get into the way of user control and freedom (3). It is a user-friendly decision to recommend a lightweight model in the beginning, which is usable for a wide variety of consumer hardware.

[29] Element Labs Inc. (2025). LM Studio. https://lmstudio.ai/

Figure 4. LM Studio comes with three modes (User, Power User, Developer) to customize the user experience. Power User is the default mode.

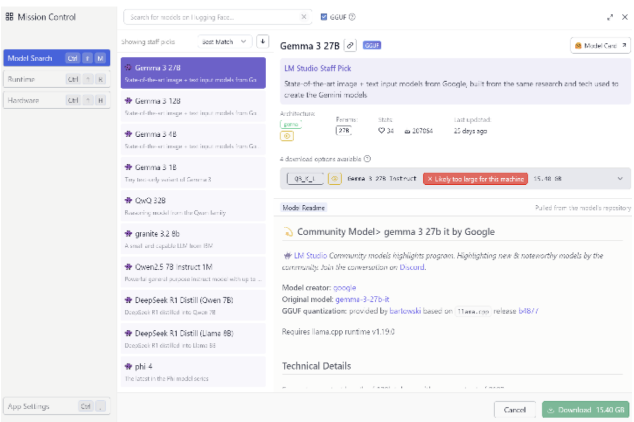
Like Jan, LM Studio includes a top navigation bar.

File	Edit	View	Window	Help
Quit Ctrl+Q	Undo Ctrl+Z	Actual Size Ctrl+0	Minimize Ctrl+M	Technical Documentation
	Redo Ctrl+Shift+Z	Zoom In Ctrl++	Zoom	LM Studio Blog
	Cut Ctrl+X	Zoom Out Ctrl+-	Close Ctrl+W	LM Studio Website
	Copy Ctrl+C	Toggle Full Screen F11		
	Paste Ctrl+V			
	Delete			
	Select All Ctrl+A			

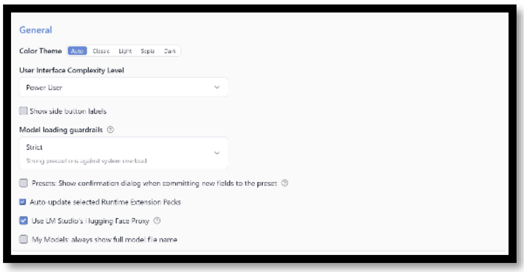
Table 9. Top navigation bar for LM Studio.

The taskbar on the left side of the client includes quick links to **Chat** scene, **Developer** scene, **My Models** scene and **Discover** scene. At the bottom, there is a link to **Downloads**, which lists history for all the downloaded models.

Interestingly, this taskbar is only visible when the client is in Power User or Developer mode. If the user switches to User mode, the '*advanced configuration*' button ( ) in the chat scene (which is presumably what power users and developers would use) changes to a search button ( ) that takes the user to the model search window, effectively being replaced with what would otherwise be Discover link in the taskbar.



The model selection window is called "Mission Control", and lists models directly from HuggingFace. Mission Control also includes settings for Runtime Extension Packs and Hardware information (System Resources). Hardware view will include details whether the hardware is compatible for the LLM tasks.



For flexibility and ease of use (7), LM Studio's Mission Control (which is the same window where models are searched and downloaded) includes app settings. From general settings, the user can change the color theme of the application.

By default, color theme is set as "auto" and therefore follows the operating system setting.

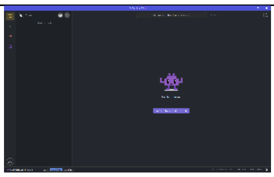
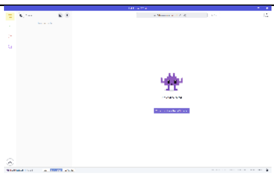
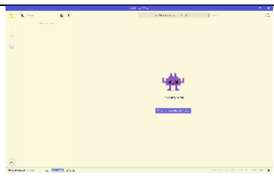
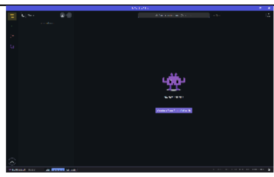
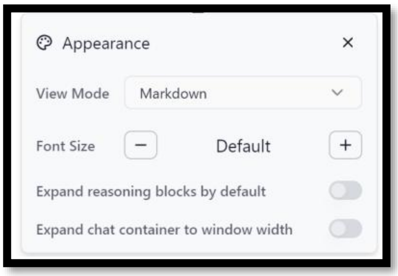
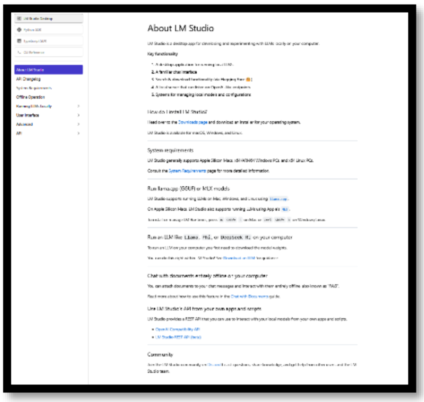
LM Studio's color themes			
Classic	Light	Sepia	Dark
			

Table 10. Four customization options for LM Studio.



In addition to color theme settings, there are appearance settings within the chat interface, where the user can change the text view mode (Markdown, Plaintext, Monospace), font size of the prompt text and the ability to expand reasoning blocks by default and expand chat container to window width.



LM Studio includes documentation (10).

The documentation is split into 8 sections: **About LM Studio**, **API Changelog**, **System Requirements**, **Offline Operation**, **Running LLMs locally** (Overview, Manage chats, Download an LLM, Chat with Documents, Import Models, Config Presets), **User Interface** (Languages, UI Modes, Color Themes), **Advanced** (Per-model Defaults, Prompot Template, Speculative Decoding) and **API** (Overview, Headless Mode, Idle TTL and Auto-Evict, Scructured Output, Tools and Function Calling, REST Endpoints) .

These are only for LM Studio Desktop – the site also includes separate documentation for LM Studio Python SDK, LM Studio TypeScript SDK and LM Studio's CLI reference.

LM Studio is available for Windows (x86 and arm64), macOS and Linux (Linux as *AppImage*).

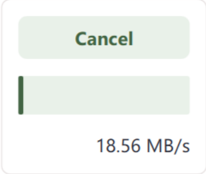
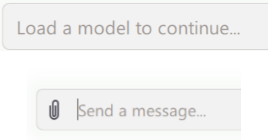
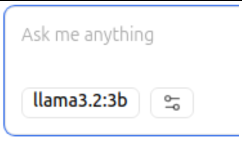
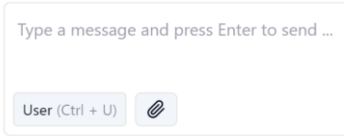
Visibility of System Status (downloading the model)		
 GPT4All	 Jan	 LM Studio
Match Between the System and the Real World (speaks user's language)		
 GPT4All	 Jan	 LM Studio
User Control and Freedom (prompt control)		
 GPT4All	 Jan	 LM Studio

Table 11. Usability heuristics between three LLM clients.

All the covered clients are functional for daily use and based on closer inspection, differences between the clients often would relate to the user's personal preferences. Same (or similar) features can be discovered from all clients, but the overall implementation differs slightly between them. In addition, specific target audiences may also vary and this, for instance, reflects in LM Studio's 'Power user' and 'Developer' features. However, significant differences in the core features most users of local LLMs would likely be interested in are relatively minimal, and all the clients can provide a proper environment for running LLMs locally. For example, all three covered clients allow the users to upload files.

Despite the most common file types such as PDFs being generally supported, there may be some differences between the support for less common file types. Whereas Jan and LM Studio mention specific file types that would set direct use case restrictions, GPT4All's LocalDocs feature uses on-device embedding models which index the folder into text snippets. Theoretically this means that the user can add more file types in the settings, but the less default file types have not been tested, and issues may arise from using them. Because there are differences between the file type support, it seems advisable to search for a client that has the best support for the specific use case, should the user decide to use the models for files, e.g. data handling and analysis tasks.

2.3.3 Local server deployment

In the context of local LLMs, local server deployment can be a useful addition, as it enables multiple devices to utilize the processing power of one computer to run local LLMs. In this case the computer would, in addition to running local LLMs, function as a server in the local area network (LAN). Such implementations may be ideal for enterprise use and in homes where there are multiple devices and/or multiple users.

Many commonly used local LLM tools and software often make it relatively easy to set up a local (*localhost*) server, after which the local port can be used for communication with the models. The exact ways of communicating with the LLM client/framework (back-end in this case) varies between implementations but documentation for specific client or framework should generally provide sufficient instructions for these use cases.

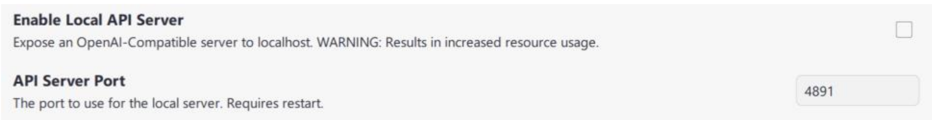
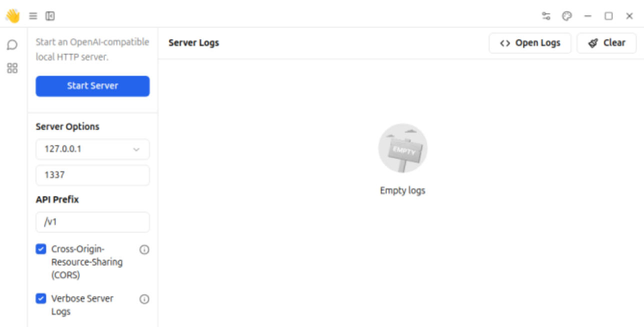
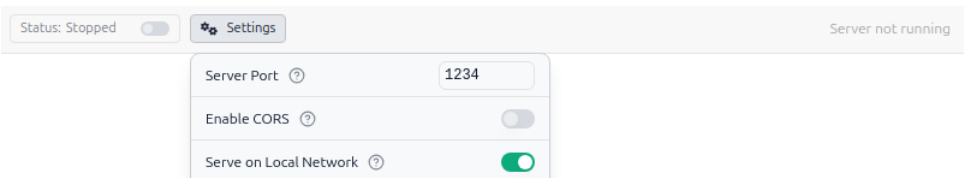
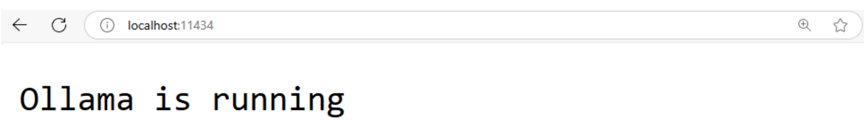
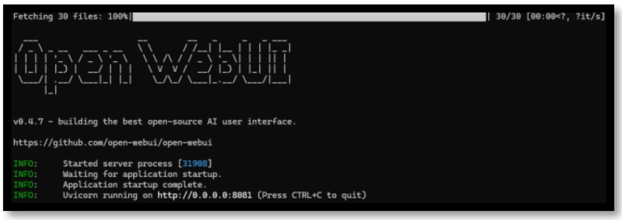
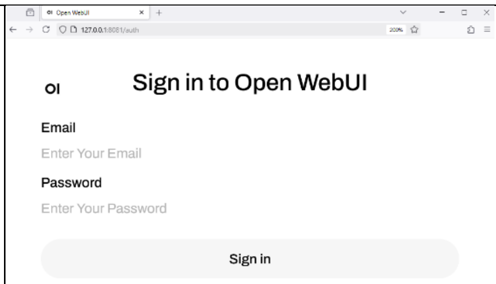
Using LLM clients for local server deployment	
GPT4All	
Jan.ai	
LM Studio	
Ollama framework running on the device	

Table 12. Examples of local server (localhost) implementations across LLM clients. Setting up a local server is often either an automatic process (e.g. Ollama) or a simple toggle-on setting.

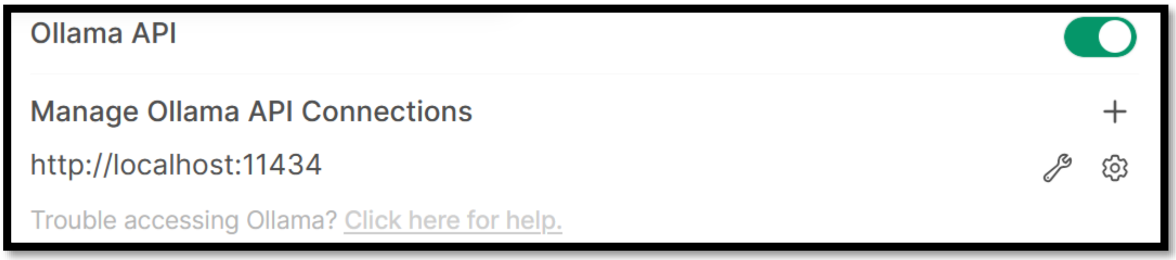
It is important to consider that hosting a local server increases the use of resources and computers may be unable to process multiple prompts simultaneously in cases where there are multiple simultaneous users, especially with home computers running larger models. In these circumstances, the processing of the prompts would typically be queued, resulting in wait times. In enterprise use, these implementations can operate on more optimized and powerful server hardware. Additionally, with certain methods, such as home network VPN or port forwarding, it is possible to make servers accessible from outside the home. That considered, with any servers that expose ports to other devices (either on LAN or on the internet), the server administrators should consider proper security measures of these implementations when needed such as encrypting this network traffic. Setting up the local server, at minimum, deploys the back end. Front end, such as the user interface (UI) to communicate with the back end (server), may have to be deployed separately if it is not included. Front end deployment for local servers provides additional benefits, as many interfaces (such as Open WebUI) include account database, admin panel and several additional features to enhance the overall usability.

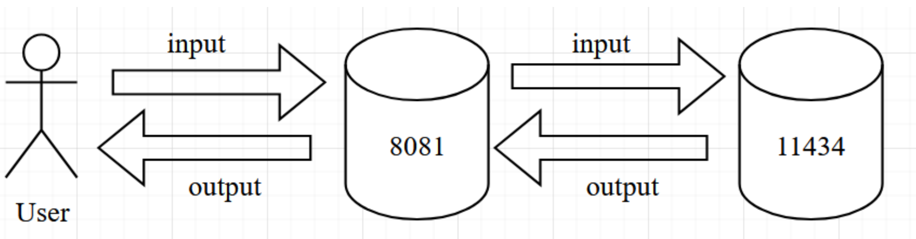


Front end (Open WebUI[30]) is being loaded. Running on port 8081.



localhost:8081 being accessed.





Visualized with draw.io[31]

Table 13. User communicates with the front end (localhost:8081), front end communicates with the LLM back end (localhost:11434). By opening port 8081, other devices can access the front end on the network, which can access the LLM back end.

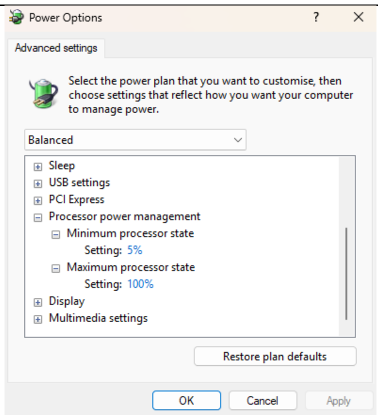
[30] Baek, T. (2025). open-webui: User-friendly AI Interface. GitHub. https://github.com/open-webui/open-webui
[31] JGraph. (2025). diagrams.net, draw.io. https://www.diagrams.net/

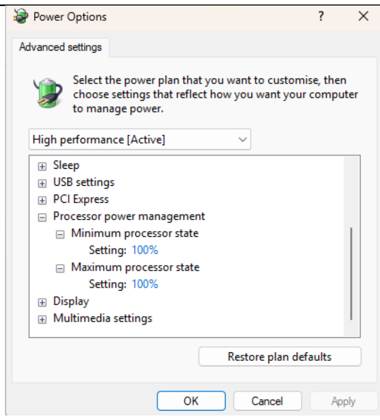
2.4 Measuring hardware requirements

2.4.1 Processing power

Central Processing Units

Running local LLMs with central processing units (CPUs) is sensitive to single-core speeds and due to this factor, higher clock speeds are preferred, as this improves the single-core performance. CPU speeds are often divided into two categories: base clock and boost clock. If the computer's operating system is running on balanced mode, the CPU will operate with base clock speed and gradually increase towards boost clock when needed. If the system is running on high performance, it will always operate with boost clock speed. Whenever local LLMs are being run for longer periods of time, it may be advisable to keep the system on high performance mode to ensure that performance remains optimal throughout the session without disruption.

<table>
<tr><th>Balanced (min 5%, max 100%)</th><th>High performance (min 100%, max 100%)</th></tr>
<tr><td>


</td><td>


</td></tr>
</table>

Table 14. Windows Power settings with minimum and maximum processor state (default settings).

Smaller models, ranging from 1 to 8 billion parameters, are typically more suitable for consumer CPUs. The core count is also an important factor when CPUs are used since many LLM clients utilize multi-threading capabilities for efficient performance optimization, which means that the overall workload is distributed across multiple cores. The exact extent of multithread processing varies between clients (e.g. some clients use 2-4 cores, and some may use more than 4), but multithread processing is, nonetheless, a standard practice in handling CPU-demanding tasks, such as running local LLMs. Therefore, fast per-core performance combined with enough cores for the task is typically needed to accomplish optimal performance when CPUs are being utilized.

Graphics Processing Units

Graphics Processing Units (GPUs) offer significant advantages over CPUs for running local LLMs. Whereas CPUs consist of limited number of cores, modern GPUs contain hundreds or thousands of cores. These cores are also “smaller”, making them more practical for AI-specific (e.g. machine and deep learning) tasks such as vector operations or matrix calculations. In short, GPUs are optimized for high parallelism with a high number of cores, whereas CPUs are optimized for few complex threads with limited number of cores. This simple distinction in architecture makes the performance difference between CPUs and GPUs significant.

The description above only applies to dedicated GPUs, which are specifically built for high-performance usage such as graphics design or gaming. In contrast, integrated GPUs are embedded into the CPU, which severely limits their complexity and the number of cores that they have. Whereas dedicated GPUs have hundreds or thousands of cores, integrated GPUs typically have only few cores and therefore miss this high parallelism capability that dedicated GPUs have.

Users who wish to use dedicated GPUs for running local LLMs should pay attention to having enough video memory (VRAM) for the model that they wish to run and high memory bandwidth. Memory bandwidth refers to the transfer speed between VRAM and GPU cores. Generally, most modern GPUs (e.g. NVIDIA RTX 3000 and 4000 series) will meet the demands for bandwidth to be sufficient for regular LLM usage. The more restricting factor tends to be the amount of free VRAM.

Graphics Processing Unit	
Memory bandwidth *Speed*	Video memory (VRAM) *Size*
Determines how quickly data flows between VRAM and GPU cores. Generally, more bandwidth means faster operation.	Determines how large LLM can be loaded into the GPU without speed-degrading operations (e.g. memory swapping).
Typical speed for dedicated modern consumer GPUs is hundreds of GB/s.	Typical VRAM for dedicated modern consumer GPUs is from 8GB to 24GB.

Table 15. GPU considerations for running local LLMs.

Neural Processing Units

Local LLMs can be run with neural processing units (NPUs), replacing CPU or GPU usage. This would be beneficial for portable devices, as consumer NPUs are tailored for AI tasks and more common in laptops than desktop computers (as of April 2025) and can provide good performance in space-efficient manner. However, due to the different architecture from CPUs and GPUs, the deployment of LLMs for NPUs can be a rather time-consuming process, as the models would have to be specifically recompiled for NPU usage.

The amount of required time for recompiling will depend on the goal of optimization that is set for the usage, e.g. deeper optimization that achieves maximum performance from the NPU will be more time-consuming. Close cooperation with NPU vendors, such as Qualcomm and Intel, can enable access to optimized tools and solutions that may further simplify the recompiling processes for AI developers.

In general, the speed of the NPU adaptation will largely depend on how the public adapts to NPUs and whether the distribution of the chips will broaden overtime, from limited series of AI-laptops to other consumer laptops and desktop computers. More specifically, NPUs must become mainstream instead of being seen as special products for special use cases. If NPUs are being sold in "exclusive AI PCs" rather than being included in regular computers by default, there is a reasonable risk that the general adaptation will remain small. In return, AI developers would likely not see the implementation of NPU support to be worth the effort as the significantly larger userbase would still use GPUs for LLMs instead.

As of April 2025, there are few LLM clients out there that support NPU utilization for limited number of models. AnythingLLM, developed by Mintplex Labs, has local AI optimization for NPUs for Snapdragon X Series Devices[32]. The library (Qualcomm QNN) currently includes models such as Llama 3.2-3B and Llama 3.1-8B – both of which can be used with consumer laptops, with 8-billion parameter model having higher hardware requirements than the 3-billion parameter counterpart.

---

[32] Aggarwal, D. (2025). Porting AnythingLLM to Windows on Snapdragon. Qualcomm. https://www.qualcomm.com/developer/blog/2025/01/porting-anythingllm-npu-windows-on-snapdragon

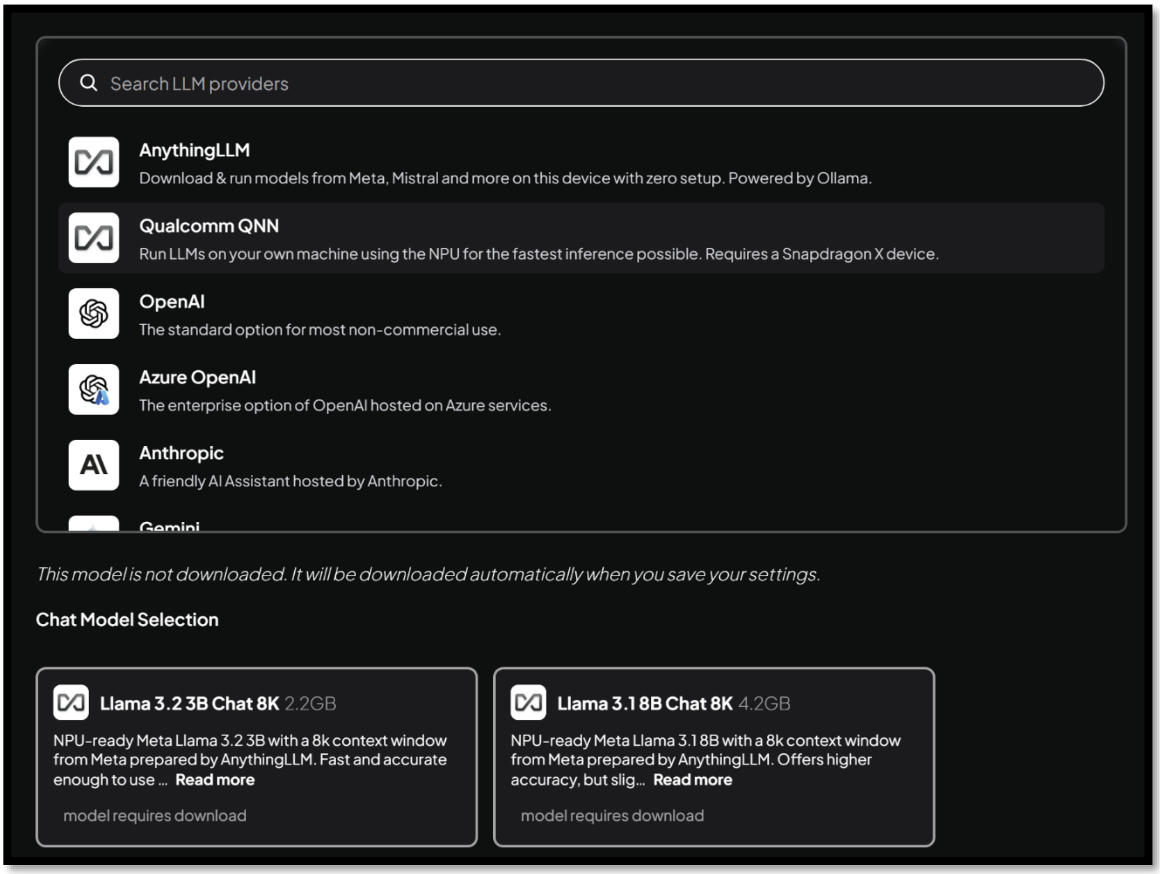


Figure 5. NPU support for local LLMs is available in some LLM clients, such as AnythingLLM[33]. The number of models supporting NPU utilization is still relatively limited.

Due to the optimization for AI/ML workloads, the processing power of current NPUs would be sufficient for running all the commonly used models that would enable good capabilities for general home and enterprise use cases. This considered, it is likely that the limitations in NPU-based home devices would be related to memory requirements rather than the processing power of the NPUs, given that the models would be recompiled in a manner where they take full advantage of the available processing power of the NPUs.

2.4.2 Memory requirements

Memory requirements for running LLMs are an existing challenge, regardless of whether the models are run with central processing units (CPUs), graphics processing units (GPUs) or neural processing units (NPUs). Because of the limited availability of NPUs in regular computer configurations, currently the most common and standardized approach is to run local LLMs with GPUs for maximum efficiency, as GPUs are more optimized for the tasks than CPUs. In typical use case, the model would be loaded into random access memory (RAM) and then the

[33] Mintplex Labs Inc. (2025). anything-llm. GitHub. https://github.com/Mintplex-Labs/anything-llm

model weights will be allocated from RAM to GPU for use. In hardware configuration where the GPU is dedicated, either the entire model or parts of it are loaded from RAM to video memory (VRAM). In contrast, integrated GPUs would use shared VRAM, where GPU would essentially use RAM instead. Dedicated VRAM is noticeably faster than RAM and thus, shared VRAM. However, dedicated GPUs may also use shared VRAM, but this should only occur when dedicated VRAM is in full use.

When speed is relevant, it is advisable to measure the available dedicated VRAM in comparison to the model size. Once the shared VRAM (RAM) is being used for the tasks, the model will not perform as efficiently as it is then partially being used from the "slower" memory.

To measure memory requirements for running the models and to draw a broader overall understanding, it is crucial to understand the optimization methods for LLMs. A common optimization method for modern LLMs is quantization, which reduces the overall memory requirements. Depending on technique used, model sizes can be reduced without significant quality impact. For example, findings by Jin et al. (2024) suggest that 4-bit quantization can maintain performance which is comparable to the non-quantized counterparts[34].

The names of the quantization techniques (e.g. Q1, Q2, Q3 and Q4) refer to the number of bits used per parameter. It is possible to recognize the absolute boundaries for running the models on a particular hardware setting by knowing their specific quantization techniques.

1-bit quantization (Q1)	**2-bit quantization (Q2)**	**3-bit quantization (Q3)**	**4-bit quantization (Q4)**
$\frac{\text{Available memory } \textit{in bytes}}{0.125 \text{ bytes } (1 \text{ bit})}$	$\frac{\text{Available memory } \textit{in bytes}}{0.25 \text{ bytes } (2 \text{ bits})}$	$\frac{\text{Available memory } \textit{in bytes}}{0.375 \text{ bytes } (3 \text{ bits})}$	$\frac{\text{Available memory } \textit{in bytes}}{0.5 \text{ bytes } (4 \text{ bits})}$

Table 16. 1-bit, 2-bit, 3-bit and 4-bit quantization methods, where each bit represents 0.125 bytes. Dividing available memory (in bytes) with the byte count results in maximum number of parameters that can be loaded into the memory.

In accordance with table 16, in a hypothetical situation where the end-user would have a Q4-based model with 8 GB (8,589,934,592 bytes) of free memory in use, the absolute maximum

[34] Jin, R., Du, J., Huang, W., Liu, W., Luan, J., Wang, B., & Xiong, D. (2024, August). A comprehensive evaluation of quantization strategies for large language models. In Findings of the Association for Computational Linguistics ACL 2024 (pp. 12186-12215).

the user could run would be 8,589,934,592 / 0.5 = 17 179 869 184 (~17.1 billion) parameters, given that all the contents would be always loaded into this memory without, for example, swapping between the disk and the memory. The precise amount of free memory for the model is going to vary as the LLM client or framework, the operating system (OS) and background processes would already reserve some amount of memory for themselves on the background. To do more precise evaluation, the memory requirements of such external processes would have to be taken into consideration when calculating the maximum memory capabilities of any computer for running the models, assuming that these processes would be running on the background. For example, a typical modern Windows home installation would already require 2-4 GB of free RAM for its operation alone, depending on how the system is configured. Therefore, the overall RAM requirement for the operating system alone would already be equivalent to running 4-8 billion parameters on Q4. Several industry-leading model series such as Llama and Gemma often utilize 4-bit quantization, as it offers reasonable balance between performance and efficiency. More "aggressive" quantization, such as Q1, Q2 or Q3, would enable even more parameters for the available memory but results in worse quality.

Apart from operating system requirements and various background processes, the overall memory requirements are also dependent on several other factors, such as whether the entire model is loaded into VRAM or only parts of it and whether the copy that exists in RAM is released or kept after the loading into VRAM is completed. In cases where the 17-billion parameter model that uses 4-bit quantization is loaded in its entirety into VRAM and not released from RAM simultaneously, this means that loading the model into memory will require approximately both 8GB of RAM and VRAM (and thus 16GB of free memory in total), essentially doubling the memory requirements. Therefore, the overall memory requirements are also dependent on the client or framework, which ultimately directs how the models are being run and optimized for usage. For instance, model offloading refers to a common method where only parts of the model are loaded into VRAM (Joo et al., 2024)[35]. Such methods may effectively reduce the overall memory requirements.

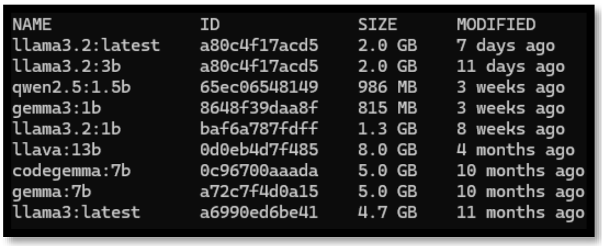
Earlier description only demonstrates the absolute maximum that could be loaded into the memory. However, it is important to note that the absolute maximum is not equivalent to the most optimal size for the hardware configuration and heavier model is still going to perform

[35] Joo, D., Hadidi, R., Feizi, S., & Asgari, B. (2024). Endor: Hardware-Friendly Sparse Format for Offloaded LLM Inference. *arXiv preprint arXiv:2406.11674*.

slower than a lighter one. Despite this, understanding the absolute maximum helps avoid the most unideal circumstances and eases the recognition of the optimal memory recommendations for models once other factors are also considered.

2.4.3 Disk requirements

Depending on the size of the model and the quantization technique, a typical model for home usage (e.g. from 1 to 8 billion parameters) is going to require approximately 1 to 5 gigabytes of storage space. Further storage space is needed if the user wants to download and use multiple models on their device.



NAME	ID	SIZE	MODIFIED
llama3.2:latest	a80c4f17acd5	2.0 GB	7 days ago
llama3.2:3b	a80c4f17acd5	2.0 GB	11 days ago
qwen2.5:1.5b	65ec06548149	986 MB	3 weeks ago
gemma3:1b	8648f39daa8f	815 MB	3 weeks ago
llama3.2:1b	baf6a787fdff	1.3 GB	8 weeks ago
llava:13b	0d0eb4d7f485	8.0 GB	4 months ago
codegemma:7b	0c96700aaada	5.0 GB	10 months ago
gemma:7b	a72c7f4d0a15	5.0 GB	10 months ago
llama3:latest	a6990ed6be41	4.7 GB	11 months ago

Figure 6. An example of Ollama[36] model list query, listing installed models on the device. The total disk space needed for 9 listed models is *approximately* 30 GB. Average size per model is 3.3 GB.

While storage space requirements are generally quite moderate among the smaller models, the hard drive type is an important factor to achieve optimal model performance. Since solid state drives (SSDs) are significantly faster than traditional hard disk drives (HDDs), loading the model from SSD to memory is faster than HDD. Due to this improvement in performance, internal HDDs are typically not found from modern computers anymore, as SSDs are strongly recommended for optimal performance of modern operating systems, such as Windows 11. However, since HDDs are more affordable storage space compared to SSDs, they are still relatively common for backups and larger storages. While storing models on HDDs can be a good approach, running them directly from HDDs results in speed degradation.

2.5 Model evaluation methods

2.5.1 Objectivity and subjectivity

When evaluating models, separation of objectivity from subjectivity is important. The answers that the models provide are not always solely objective, as they may include opinions on various issues. Despite this, typically the model would have a valid reasoning for the provided opinions,

[36] Ollama. (2025). Ollama. GitHub. https://github.com/ollama/ollama

where objective data was used to reach a subjective conclusion. The user having a conversation with the model, however, may disagree with the conclusions that the model provides based on the objective data. This may happen even if the user does not disagree with the objective data. Evaluating the model can therefore be divided into three (3) categories.

Objective answers only	Subjective answers only	Objective and subjective answers
The questions provided to the model need to be direct and there exists an answer to them that is uncontroversial. An example of such a question would be a mathematical one.	The questions provided to the model provide more room for answers and philosophical thinking. The question may start with words such as "what should" or "what do you think".	A combination of objective and subjective questions, which can be either separated or embedded into the same input. Hybrid approach to the evaluation may be needed.

Table 17. Three-category division for model evaluation.

This separation is important to acknowledge, as the end-user may value model's capabilities to answer questions either objectively or subjectively. They may expect both capabilities from the model. Therefore, it is possible to measure either one of the two capabilities or both. However, subjective answers, to some extent, lead to a new evaluation problem. Because they are subjective in nature, it may be difficult or even impossible to evaluate them objectively. For example, if the model performs a subjective task such as creative writing (e.g. how the story *should* continue), it is up to those evaluating the model to form a more subjective view as to how creative the model was with the story generation. This presents obvious challenges, as story that is creative in one person's opinion may not be creative to another person.

In theory, an argument could be formed that it is not possible to evaluate such subjective answers objectively. Rather, the goal should be to be *as objective as possible*. The evaluation methods, such as benchmarks and user studies, can cover both the objective and subjective sides of the models on a case-by-case basis. For example, there are many benchmarks consisting of scientifically validated questions with objective answers and in contrast to those, there are benchmarks for evaluating more subjective capabilities such as emotional intelligence and creative writing, though latter ones have been occasionally questioned as these approaches tend to introduce further challenges.

Due to the complexity of evaluating subjective tasks, many frameworks have been introduced to address these existing challenges. For example, Zhao et al. (2024) introduced Language Model Council (LMC) for highly subjective tasks, which is a democratic three-step process where a test set is formulated through equal participation (1), administered among council members (2) and responses are evaluated as a collective jury (3). In the LMC approach, LLMs operate as juries for the provided responses, automating the evaluation procedure similarly to other benchmarks. The study found that LMC correlates more with rankings established by human judges than other benchmarks.[37] Subjective answers may be evaluated by human judges or other approaches, such as LMC, may be introduced instead. The major point is consensus-driven evaluation which will, to the extent possible, reduce subjective biases from this process.

2.5.2 Standardized benchmarks

Benchmarks are a set of tasks that aim to evaluate the model's performance. In benchmark evaluation, each compared model is put through the same tasks and based on the overall performance, they receive a score in the end. Scores are then compared to form a scoreboard, which will serve as a baseline for users to discover models that best suit their use cases. For example, whereas one model may be good with textual analysis and language understanding, another one may be best suited for programming, e.g. writing code or correcting syntax errors. The process of benchmarking is typically automated without human intervention, especially with tasks that are reliant on computational capabilities, i.e. objective tasks. However, human insight may occasionally be needed for proper evaluation once the context of the task leads to a response that is more subjective and open-ended.

There are differences between the implementations of benchmarks and some of the benchmarks may also be academically validated works. For instance, Rein et al. (2024) presented A Graduate-Level Google-Proof Q&A Benchmark (GPQA), which consists of 448 multiple-choice questions written by domain experts. A dataset analysis was conducted to ensure that the questions are extremely difficult and of high quality, detailing both question objectivity and question difficulty to ensure that the multiple-choice questions are challenging even for domain experts who have unlimited access to the web during the testing process, hence this benchmark being referred to as “Google-proof”.[38]

Benchmarks are generally efficient, as they can be used to evaluate *thousands* of different models through *thousands* of different tasks, often as fully automated process. However, the

---

[37] Zhao, J., Plaza-del-Arco, F. M., & Cercas Curry, A. (2024). Language Model Council: Benchmarking Foundation Models on Highly Subjective Tasks by Consensus. arXiv e-prints, arXiv-2406.
[38] Rein, D., Hou, B. L., Stickland, A. C., Petty, J., Pang, R. Y., Dirani, J., ... & Bowman, S. R. (2024). Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

reliability of benchmarks as an evaluation method is occasionally debated. Benchmark chasing refers to the practice where models are specifically trained to perform well on the benchmarks. This practice is considered particularly challenging from evaluation perspective because creators of the models may achieve better results by implementing tricks rather than concrete improvements in the model quality that would accurately correlate with the increase in the scoring. As a result, this may negatively impact the overall relevance of the benchmark results. (Woodside & Toner, 2024)[39] How often and to which extent benchmark chasing truly happens is likewise debated. Considering that scoring well on benchmarks is a method to draw attention and a wider marketing advantage for the models, it is not out of the question that benchmark chasing would, in theory, be in the interests of the AI companies that are closely competing with one another.

In addition to the challenges of benchmark chasing, benchmarks are somewhat sensitive to changes, and this may reflect in the leaderboard scoring. For instance, Alzahrani et al. (2024) demonstrated in a study that changing the order of choices or the method of answer selection resulted in rankings changing up to 8 positions in multiple-choice question benchmarks, such as MMLU[40]. The findings may question the idea that benchmarks should be taken at face value when LLMs are evaluated. Of course, this criticism does not mean that benchmarks for LLMs ought to be rejected. In fact, they are still good at evaluating LLMs in a variety of fields and getting a more general idea of the model performance. Rule of the thumb being that if a model does not perform well, in general, it is not going to receive a good score in the benchmarks either, regardless of factors like the order or benchmark chasing. Merely, these findings demonstrate that once a certain threshold is passed in the quality of the LLMs, and the gap between good and bad performance becomes less narrow, the capabilities to evaluate the models accurately and objectively narrow down as well. Whether one model outperforms another may be based on very minor specifics in the measurements.

Lastly, benchmarks face criticism for having lack of wide applicability, raising the question if they truly represent real-world use cases as broadly as would be desired. Benchmarks often tend to cover fields which are already within the general focus scope of the LLM training. However, the broad range of general use cases in real-world environments may go beyond such scopes.

---

[39] Thomas Woodside and Helen Toner: Evaluating Large Language Models. CSET | Center for Security and Emerging Technology. July 17, 2024. https://cset.georgetown.edu/article/evaluating-large-language-models/

[40] Alzahrani, N., Alyahya, H. A., Alnumay, Y., Alrashed, S., Alsubaie, S., Almushaykeh, Y., ... & Khan, H. (2024). When benchmarks are targets: Revealing the sensitivity of large language model leaderboards. arXiv preprint arXiv:2402.01781.

Regardless, using benchmarks is a good approach for evaluating models overall if the potential shortcomings are being acknowledged.

2.5.3 User studies

In addition to benchmarks, studies may be conducted directly with people who use the models. This may be done in a variety of ways. For instance, many LLMs are also usable in a form of an online service. Typically, when the user receives a response in the online service, there is an option to rate these answers.

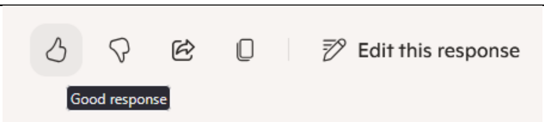
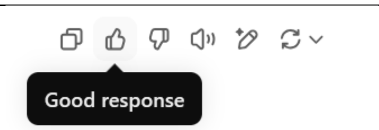
Rating systems for online LLMs	
Microsoft Copilot[41] https://copilot.microsoft.com	OpenAI ChatGPT[42] https://chatgpt.com/
	

Table 18. Various LLM web services allow users to rate the outputs.

Although these rating systems are generally very simplistic, such as thumbs *up* or *down* in table 18, this rating functionality provides immediate feedback to the developers and is used for statistical analysis to understand how well the model performs in different types of fields and to prioritize areas where further improvements are most needed. However, in the context of local LLMs, some of which may be entirely tailored for offline use, these immediate feedback systems would not exist. In these kinds of cases, user (usability) testing within offline environments may be conducted.

In addition to feedback provided directly to the model providers as showcased in table 18, the models can also compete in this regard. For instance, Chatbot Arena is a commonly used online service, in which model responses to human-generated prompts are actively compared to form scoreboards[43]. Essentially, the user will type a prompt and get two responses and evaluate which response out of the two is better.

---

[41] Microsoft. (2025). Copilot [Large language model]. https://copilot.microsoft.com
[42] OpenAI. (2025). ChatGPT [Large language model]. https://chat.openai.com
[43] LMSYS Org (2025). LMArena [web service]. https://lmarena.ai/

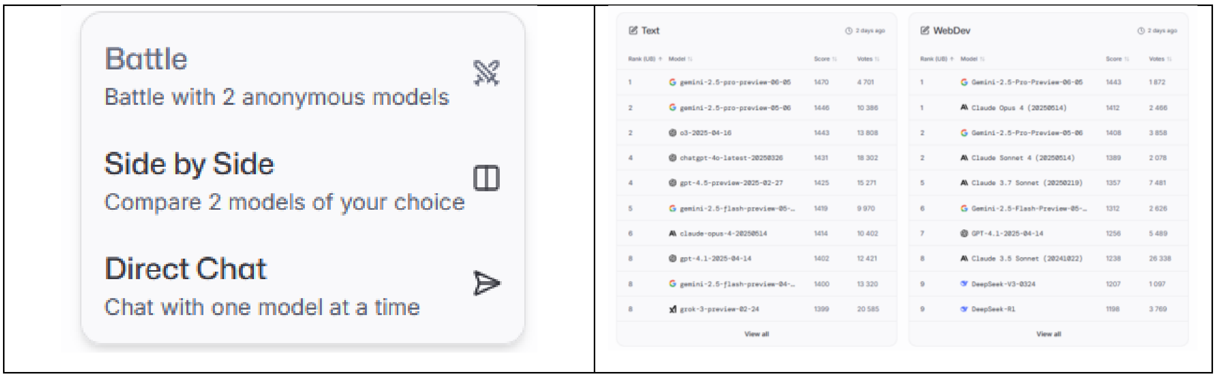


Table 19. Chatbot Arena allows its users to battle with two anonymous models, compare two models of their choosing or directly chat with one model at a time. The results by the users are then compared, and scoreboards get created.

Chatbot Arena is a relatively new concept, created in 2023. The service enables broader user studies to be conducted on a wide scale and enables models to be put into tests where real-life use cases, by real users, may be considered, as opposed to standardized benchmarks. However, despite the efficacy of directly comparing models in more user-oriented use cases, Chatbot Arena has also received criticism and several improvement recommendations. For example, Singh et al. (2025) point out many factors that may lead to distorted Arena rankings, such as Chatbot Arena not having a requirement for all models to be made public upon submission and the potential of several major model developers benefiting from private, extensive testing[44]. In this context, it is important that the processes remain transparent, and the community is at the forefront of the studies.

User studies, more broadly, can be divided into two categories: they can be conducted in real-life environments or in controlled environments. Real-life environments would include people using the models for real tasks that they are trying to complete. Controlled environments would include a set of tasks that would be given to the user to complete, or they would create the tasks themselves, specifically for the purposes of only testing the model. In the wider context of software design and development, both user study approaches are typically desired. For instance, usability testing of a piece of software is typically being conducted in controlled environments before software is released. After software is released, data is being collected from users who are using the software in real life, such as in a form of telemetry and analytics.

[44] Singh, S., Nan, Y., Wang, A., D'Souza, D., Kapoor, S., Üstün, A., ... & Hooker, S. (2025). The leaderboard illusion. arXiv preprint arXiv:2504.20879.

In comparison to telemetry and analytics, which are often relatively straightforward implementations with the existing web frameworks, setting up a controlled test environment requires more careful planning and approach. For instance, separate test protocols need to be designed and created and participants recruited. Each participant will then complete the test in accordance with the protocol. In addition, test moderators (known as facilitators) and observers may sometimes be needed to conduct the testing, based on the objectives of the study. Thus, controlled user studies, despite being helpful in user-centric design and evaluating broader usability, can be time-consuming to set up and conduct.

2.6 Challenges

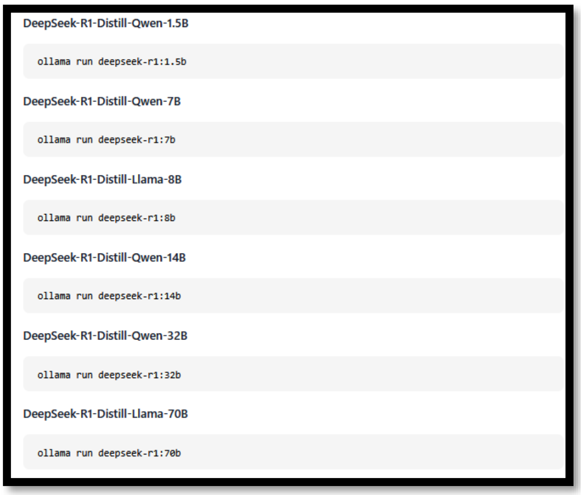
2.6.1 Knowledge cutoff

A widely acknowledged disadvantage of LLMs is the knowledge cutoff. Knowledge cutoff refers to a phenomenon in which LLMs are incapable of knowing certain details such as events that are tied to specific times. This is due to training data being from a period before such events took place. The disadvantage in question can be noticed within the context of local LLMs in cases where the internet is not accessed and therefore creates expectations that the models need to be updated, or new models would come out sooner or later which include more recent information. However, the training processes of LLMs are time-consuming and sometimes very expensive. Unless the internet is being accessed, knowledge cutoff may be an inevitable disadvantage. However, it may be reasonable to treat local LLMs similarly to physical books in this regard, where the knowledge cutoff may be seen to be the publication date of the book. Just like the person would get more up-to-date information by purchasing a newer book, they would install a newer LLM once it becomes available to update the knowledge base.

This challenge is further emphasized by the fact that the common way of addressing the knowledge cutoff phenomenon of the models, integration of live web search, would transform fully local LLMs to mainly local LLMs (as demonstrated in this paper as four-level categorization in figure 2). Therefore, such an approach to this challenge would be out of question for those who want to use a fully local implementation. For fully local LLMs, knowledge cutoff is likely an inevitable challenge, although faster training process and delivery with advanced technologies might, in the future, mitigate the severity of this phenomenon.

2.6.2 Choice overload concerns

Whenever users of technologies are required to choose between options that closely resemble each other, there is a concern that the amount of choice provided to them is going to have a negative impact on the wider adoption of the said technologies. This problem may intensify when each option essentially promises equally good performance. As a result, the user may feel hesitated to proceed in fear that they will make the wrong choice. The number of models to choose from can be overwhelming. It is important to note that oftentimes the model libraries do not contain only the models released by companies such as Meta and Google, but the overall nature of "open" models will lead to a circumstance in which other companies and users will fine-tune and modify the models depending on the objectives and possibly release them.


DeepSeek-R1-Distill-Qwen-1.5B

```
ollama run deepseek-r1:1.5b
```

DeepSeek-R1-Distill-Qwen-7B

```
ollama run deepseek-r1:7b
```

DeepSeek-R1-Distill-Llama-8B

```
ollama run deepseek-r1:8b
```

DeepSeek-R1-Distill-Qwen-14B

```
ollama run deepseek-r1:14b
```

DeepSeek-R1-Distill-Qwen-32B

```
ollama run deepseek-r1:32b
```

DeepSeek-R1-Distill-Llama-70B

```
ollama run deepseek-r1:70b
```

Figure 7. First-generation of DeepSeek series include distillation of Meta's Llama models, with the purpose of demonstrating that distilled smaller models can perform well on benchmarks.[45]

Even though there is some concern for choice overload, it is possible that choice overload is not as big of an issue as may be assumed. Scheibehenne et al. (2010) studied choice overload in a meta-analytic review and based on the results, reliable occurrence of choice overload could not be identified[46]. This, however, does not mean that people cannot be overwhelmed by choice – rather, the adverse effects from choice may be exaggerated. Choice overload may, in many cases, be a short-term temporary "irritation" that is forgotten after the choice is made. Additionally, future solutions may minimize these occurrences further. For example, the possibility of using recommendation agents to address uncertainty in decision-making has been

---

[45] deepseek-r1 | ollama. https://ollama.com/library/deepseek-r1 April 2025.
[46] Scheibehenne, B., Greifeneder, R., & Todd, P. M. (2010). Can there ever be too many options? A meta-analytic review of choice overload. Journal of consumer research, 37(3), 409-425.

studied in the context of grocery shopping (Rohden & Espartel, 2024)[47]. If it is the case that average users will be required to choose between multiple options that resemble each other, similar approaches could be considered in the marketing of AI tools, including local LLMs.

2.6.3 Trust issues

Many of the companies at the forefront of local AI development are big tech companies. This also reflects in LLM development. For example, Meta has Llama series, Google has Gemma series and Microsoft has its own Copilot ecosystem. This, in theory, means that many of the trust issues that exist towards big tech companies could transform to the local AI implementations, creating further challenges for wider adoption. In fact, there are examples of trust problems already occurring. One relatively common example where these trust issues can be noticed is Microsoft introducing local AI feature called 'Recall' for Windows systems. Recall, essentially, is an AI-powered program that would capture screenshots of the user's computer screen every few seconds and store them locally on their hard drive. Then, with local language models, the user could search for any piece of text or image that they have encountered in a specific timeframe. In other words, Recall would introduce photographic memory for home computers, powered by local AI capabilities. The outcome is that the computer would "remember" what it has seen before, hence the feature being called Recall.

When Recall feature was first introduced, it was met with high scepticism. As one example of this scepticism, some people were doubtful whether everything is truly processed locally, and nothing regarding the computer usage is sent to Microsoft either directly or indirectly. Telemetry data or personalization, for example, could be considered indirect communication which is created based on capturing certain pieces of information (e.g. keywords) rather than the entire computer usage by the end-user. Even though such details may be anonymized, they may, for instance, be used to serve targeted advertising to the end-user. Microsoft assures that they do not have access to the screenshots nor are they used for personalization or targeting of any kind. Recall is a good case example to demonstrate that trust needs to be gained for local AI features and emphasizes that this can be difficult.

The lack of confidence can create significant challenges for the integration of local AI features on a wider scale, specifically regarding the question of wider mistrust towards big tech

[47] Rohden, S. F., & Espartel, L. B. (2024). Consumer reactions to technology in retail: choice uncertainty and reduced perceived control in decisions assisted by recommendation agents. Electronic Commerce Research, 24(2), 901-923.

operators. An intelligent operating system or software has very little practical value if the users do not trust that the intelligent solutions are used in ways which are in their favour or do not compromise their privacy and security. Although Microsoft took some steps to address the bigger concerns that were raised by implementing authentication and stronger encryption solutions after security problems were identified with the previous approach to Recall[48], these efforts may be simply seen as an attempt to improve public relations after much of the damage had already occurred with the initial first announcement and impressions of the feature. It is also worth noting that in the context of Recall, the general privacy and security concerns are not limited to trusting Microsoft's role. Because Recall would save screenshots of the computer screen every few seconds, cybercriminals could target these saved screenshot collections to gain sensitive information which they could use against the victim, as someone's computer usage history, apart from possible exclusions that are manually set from the settings, could be saved on their hard drive. Considering such circumstances, as exciting as some local AI features may seem to their creators and providers, the meaningful excitement in practice or lack thereof is ultimately determined by the users and their trust in such features.

2.6.4 Language support and multilingualism

An existing challenge for local LLMs is to introduce multilingual support while maintaining model size that is sufficient for the consumer hardware. One approach, in theory, would be to introduce separate models for each language. However, in the current era where there are rising quality expectations for AI, this is a rather impractical approach because it would ultimately narrow down the training potential and thus have unbeneficial impact on the accuracy and knowledge quality of the models. For instance, there is much less general information available in Finnish than in English. With multilingual approach, the training can happen in multiple languages, and this is relevant both in the context of accuracy and quality of the models. Such an approach is good from the perspective of knowledge quality but introduces challenges.

A phenomenon in which multilingual data is causing trouble for both low-resource and high-resource languages due to factors such as limited model capacity, is sometimes referred to as the "*curse of multilinguality*". Chang, Tyler A., et al. (2023) discovered that pre-training with multilingual data results in improvements for low-resource languages but the improvement

[48] Cunningham, A. (2024). Microsoft is reworking Recall after researchers point out its security problems. Ars Technica. https://arstechnica.com/gadgets/2024/06/microsoft-makes-recall-feature-off-by-default-after-security-and-privacy-backlash/

comes to a decrease once model capacity limits are being approached, and the performance of high-resource language starts to suffer from the multilingual data.[49] It is also important to consider that user's interests to have multilingual support may not reflect equally on the interests of those who create the models. Choosing to release a model without multilingual support may be either a design choice or multilingual support may not be high on the priority list. In the case of Meta's Llama series for example, the training data and the general model language has traditionally leaned heavily towards English.

It seems reasonable to conclude that model sizes would increase because of added multilingual support. However, there does not appear to be any specific optimal parameter counts for multilingual LLMs. Therefore, the general question would be just how much the size would increase. By knowing this detail, it becomes possible to measure the challenges in integration. In the context of average consumer hardware, there are clear ending points as to when models would simply become too heavy to run. Different models, both larger and smaller ones in size, appear to have varying levels of multilingual support (*multilingual* support meaning the general understanding of different languages than English).

Multilinguality test – How do you say "I am" in Finnish?

```
C:\Users\teemu>ollama run llama3.2:1b
>>> How do you say "I am" in Finnish?
In Finnish, the word for "I am" is "sin" (informal) or "san" (formal). If you want to be more formal, you can use
the pronoun "minä".
```

```
C:\Users\teemu>ollama run qwen2.5:1.5b
>>> How do you say "I am" in Finnish?
In Finnish, the phrase "I am" is pronounced and said as "Minä on". The word order can vary slightly depending on
context but generally follows this structure: "Nimistä on nimi", where "nimi" means name. For example:
- Minä on kaksi (I am two)
- Minä olen asema (I am an object)
```

```
C:\Users\teemu>ollama run gemma3:1b
>>> How do you say "I am" in Finnish?
The most common way to say "I am" in Finnish is:

**Minä olen.**

Here's a breakdown:

*   **Minä** - I
*   **on** - am

You can also say:

*   **Olen** - I (This is a shorter, more casual way to say it)

So, **Minä olen** is the best and most versatile option.
```

Table 20. Out of three compared models (Llama3.2-1B, Qwen2.5-1.5B, Gemma3-1B), Gemma3-1B appears to have most accurate understanding of Finnish. Gemma 3 is advertised as "multilingual", which also likely reflects in the capabilities of its smaller-size (1B) variant.

[49] Chang, T. A., Arnett, C., Tu, Z., & Bergen, B. K. (2023). When is multilinguality a curse? language modeling for 250 high-and low-resource languages. arXiv preprint arXiv:2311.09205.

3 User study

3.1 Methods

3.1.1 Test participants

Six participants (age range 28-75, 3 males and 3 females) took part in the study. The pool includes a computer scientist (1), a special education teacher (2), a cardiac nurse (3), a medical researcher (4) and retired gerontologist (5) and retired physiologist (6). The participants were chosen due to their professional backgrounds and the overall interest in presenting questions and evaluating the responses by the LLM.

3.1.2 Model setting

In this study, the test participants evaluate the installed LLM (Meta Llama3-8B). The model does not use an internet connection and runs locally on a home computer. The installed LLM is run via Windows Subsystem for Linux (WSL), utilizing Ollama framework on Ubuntu. This framework is used for running the models. In addition, a user interface (UI), Open WebUI (formerly known as Ollama WebUI) is deployed for the Ollama framework. Both Ollama and Open WebUI are hosted on the same computer, with ports being 11434 for Ollama and 8080 for Open WebUI. The port (8080) from Windows host is forwarded to the WSL instance, making the front-end function on the local network, which is then used in this testing environment. Inputs and outputs are processed with a standard home GPU (NVIDIA GeForce RTX 4060 Ti, 8GB of VRAM). The model does not have access to the internet when it is being used. Therefore, the study provides a broader understanding of the type of assistance that local LLMs may offer for day-to-day questions on a standard home computer setting.

3.1.3 Interviews and data collection

Data for the study was collected between June and August of 2024 by having a sit-down session with each test participant. During the session, the participant presents four (4) main questions to the model and one (1) optional follow-up question for each main question. Participants were free to decide the fields the questions relate to and the overall formatting and style but were requested that at least one out of the four questions would, in some way, relate to their own field. In addition, they were requested to ask questions where the response could be evaluated as right or wrong, which should exclude more opinion-based (subjective) samples. The test participants asked questions they generally know answers to, therefore being able to evaluate the given responses but were also allowed to verify different parts of the information with the help of the internet during the evaluation process in case of uncertainty.

In total, there were 40 questions (24 main questions and 16 follow-up questions). Each question was presented to the model in English.

Main questions (24)	Follow-up questions (16)
What is the difference between meta-analysis and systematic review?	How broad systematic review should be?
How to become a high-quality researcher?	How long it takes to become a high-quality researcher?
Which one is better: PubMed or Medline?	Who owns Medline?
How many person aged 100 years or more lives in Finland?	What is the estimation of persons aged 100 years or more live in the whole world?
How reliable artificial intelligence is?	What is this answer based on?
Which country in Europe has the nurse highest salary?	How about Finland?
What is takotsubo?	*No follow-up question*
Urho Kekkonen as competitive athlete, which sport?	How about Urho Kekkonen and high jump?
Give me an example of 10 dumb quotes made by Finnish politicians.	Can you quote these in the original language?
Explain quantum physics to me like I am a 6-year-old child.	*No follow-up question*
Write me a short humorous rap song about Donald Trump's political aims.	*No follow-up question*
If there are 2 chickens, 2 cows, a human and a regular chair standing in a room, how many legs touch the ground?	What is the slowest way to count from 0 to 10?
What causes aggression for a little child?	*No follow-up question*
How would a responsible parent act when a child is being aggressive?	Can a responsible parent have an aggressive child?
Is there a way to get more time, and how, where?	*No follow-up question*
How can you encourage your child's curiosity?	*No follow-up question*
What are the most common medication side effects for older adults (above 75 years old)?	Are antipsychotic drugs good for those who have memory disorders?
How to define anticholinergic drugs?	What are anticholinergic symptoms?
What percentage of people above the age of 85 are in long-term care in Finland?	How about Sweden?
What are the long-term care criteria in elderly care? Who is entitled to long-term care?	*No follow-up question*
In Finland, how many percentages of 24-hour elderly care patients come back home?	*No follow-up question*
Is Lieto a city and how many inhabitants are there?	How many cows and pigs are there in Lieto? And how many animal farms?
How many sleep spindles appears in EEG to a healthy adult person?	What is the meaning of REM?
Tell me the latest new accomplishments in neurophysiology	Which one of these is the most significant?

Table 21. Questions presented by the test participants.

3.1.4 Evaluation

The test participants were requested to fill in a form after the test, evaluating the accuracy of the responses provided. This evaluation is based on Likert's 5-point scale. Value 1 is given if the response is regarded as false and 5 if the response is true. In between those two values, 2 is given if response is regarded as mainly false and 4 if mainly true. 'Mainly' means that the response is either more than 50% false or true but not entirely and was clarified in the evaluation form. Midpoint (3) is marked if the response is regarded as neutral or cannot be evaluated true or false at all. The participants also evaluate the responses of the optional follow-up questions in case they were given.

3.2 Results

Based on accuracy evaluation by the test participants, Meta Llama3-8B performed with an average accuracy rate of 73,8% for 24 main questions and 66,6% for 16 follow-up questions. The average value is calculated with true (5) indicating the score 1.00 and whereas false (1) indicating the score of 0.00. In between those two values, mainly false (2) indicates 0.25 and mainly true (4) indicates 0.75. Neutral midpoint (3) was not considered in calculating the average.

Main questions (24)		**Follow-up questions (16)**		**In total (40)**	
True (5)	41% (10/24)	True (5)	43% (7/16)	True (5)	42% (17/40)
Mainly true (4)	33% (8/24)	Mainly true (4)	25% (4/16)	Mainly true (4)	30% (12/40)
Midpoint (3)	8% (2/24)	Midpoint (3)	6% (1/16)	Midpoint (3)	7% (3/40)
Mainly false	4% (1/24)	Mainly false (2)	0% (0/16)	Mainly false (2)	2% (1/40)
False (1)	12% (3/24)	False (1)	25% (4/16)	False (1)	17% (7/40)

Table 22. Accuracy evaluation by the test participants.

When all responses are considered (as outlined in table 22) and combining main questions and follow-up questions, Meta Llama3-8B gets an overall average accuracy rate of 70,9% for total of 40 questions presented, in accordance with the method described above.

It is worth considering that even though the overall evaluation-based accuracy rate for all the responses was 70,9%, 29 out of 37 responses (78,3%) were either rated mainly true (4) or true (5) when neutral midpoints are excluded. When calculating the average, mainly true is generalized to be 75% (more than 50% true but not 100%) but test participants rating the answer as mainly true may, for example, think that the response was 60% or 90% true. The 5-point scale was used to simplify the rating process and to provide a general idea of accuracy.

During the study, there were also a few cases where mainly true was rated when the LLM provided only the right information but left something relevant unmentioned for the given question. In addition, the way the given response is organized may also have an impact on the rating. For instance, when the LLM creates a list for the answer, the order of the list could also be considered while rating the answer. This means that the accuracy rate is a combination of various factors and a broader rate for the general quality of the responses. Nevertheless, it is notable that a local, general-use LLM on consumer-end hardware gets almost 4 out of 5 of its responses rated either mainly true or true once the neutral midpoints are excluded.

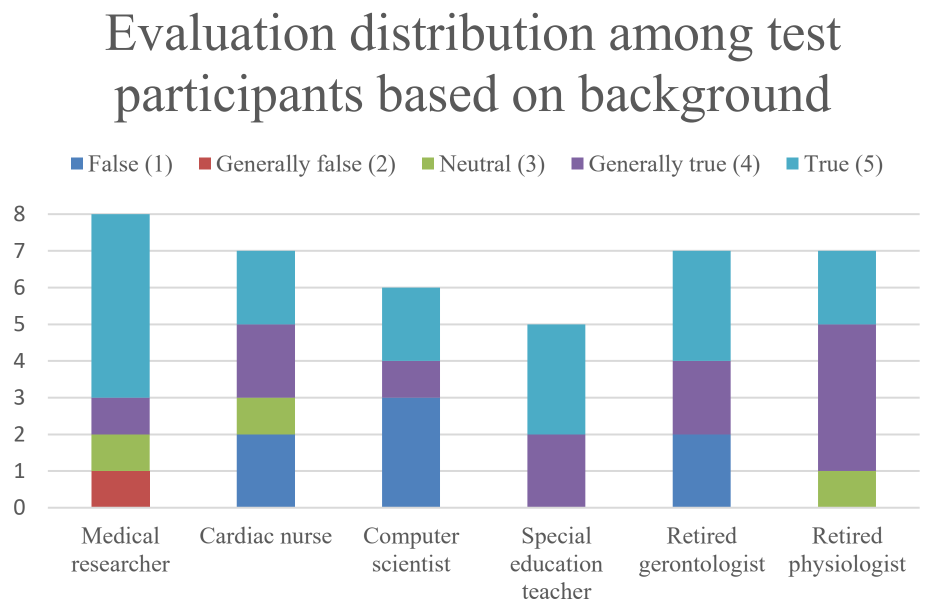


Figure 8. Evaluation distribution.

3.2.1 Challenges and speculation

While it is possible to evaluate whether the answer provided by the model is correct or not if there is an objective answer to the question, evaluating the reasons for the wrong answers given will be more challenging if not impossible and oftentimes strongly speculative. This is due to the black box phenomenon of LLMs, which makes it impossible to confirm what the output is exactly based on. One example of such speculation became apparent during the user study results. Hypothetically, it is possible for the model, based on the data it has been trained on, to connect the right dots together in wrong ways, which would then result in an incorrect response to the presented question. One instance where this could have occurred would be how the model, on one occasion, responded that former Finnish president Urho Kekkonen was an accomplished ski jumper when asked about Kekkonen's career as competitive athlete. The correct answer to this question was high jump but Kekkonen was also known for skiing frequently, although not competitively. However, based on public knowledge, Kekkonen did not ski jump at all.

The speculative approach to evaluating this response would be that the model connected *skiing* and *jump* (from high jump) since Kekkonen did both and therefore incorrectly responded that Urho Kekkonen did ski jumping. This was an incorrect response and when the model was asked a follow-up question where high jump was separately brought up, it would apologize for the mistake and correct itself.

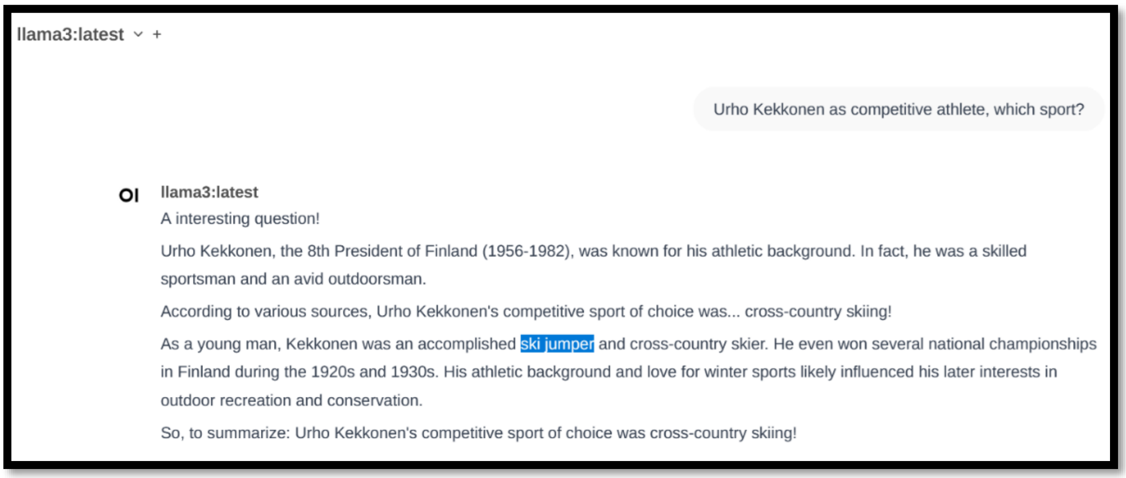


Figure 9. Llama3: "*Kekkonen was an accomplished ski jumper*"

The conflict with the response to the question presents an interesting perspective to LLM evaluation and demonstrates further challenges. In one instance, it feels encouraging to be speculative when LLMs provide such answers and try to find explanations for the answers. In another instance, it is important to keep in mind that there is no concrete way of knowing whether these speculations are at all accurate or not. Since the outputs are based on predicting the next words based on the training data, there is an ongoing process of connecting dots when the outputs are generated. In this case, it is notable that the "dots" (skiing and jump) were not incorrect, as Kekkonen did, in fact, do both. But the way they were connected when the response was being provided (generated) to the user, was. However, as previously stated, these evaluations would be highly speculative, and caution should be exercised as it is not possible to draw broader, objective conclusions from such speculations.

4 Discussion

To address the first research question of this thesis, factors hindering the usage of local LLMs are often related to memory requirements, choice overload, trust issues and lack of language (multilingual) support. Regarding the first challenge, running larger and more capable local LLMs can result in heavy processes that require a lot of memory capacity (e.g. VRAM/RAM) for proper usage, without resorting to slower techniques such as swapping between the disk and memory. This is particularly challenging as, in comparison to the memory requirements, the processing power of the components has improved significantly and therefore limited memory creates more significant barriers in this regard. Nonetheless, it is worth noting that memory capabilities in consumer devices continue to increase. In 2024, 16GB of RAM was seen as a new standard for Windows PCs[50], which is a notable increase in comparison to the previous years. Some analysts have witnessed that the average laptop RAM quota increases approximately *12%* every year[51]. In addition, given that Microsoft, together hardware manufacturers have expressed interest in "AI PCs", it is reasonable to expect that the overall memory requirements for local AI features will be also considered in this context. Therefore, there is reason to be optimistic that the most significant memory-related challenges in consumer devices will ease in the upcoming years. However, until these steps are taken more broadly, average computer configurations will be more limited to LLMs of smaller sizes. Such steps will also be effective in addressing the second research question of the thesis, which is about making local LLMs generally more accessible, although most relevant factors regarding accessibility relate to LLM clients with proper documentation and the overall usability of these solutions. The heuristic evaluation conducted in this paper demonstrates that the common clients for local LLMs are already quite usable and successful efforts have been made to create efficient *one-click* access to the models. Further work around the solutions and continued usability enhancements are important.

Choice overload is a phenomenon that can hinder the adoption of new technologies more generally, not only being limited to AI features and models. Although choice can certainly be a positive factor for some consumers, many of them can be overwhelmed by it. However, it is

[50] Batt, S. (2024). Why 16GB of RAM Is the New Standard for Windows PCs. MUO. https://www.makeuseof.com/16gb-ram-new-standard-windows/
[51] Morales, J. (2024). Analysts say average laptop RAM quota will reach 11.8GB in 2024 — up 12% year-on-year. Tom's Hardware. https://www.tomshardware.com/laptops/analysts-say-average-laptop-ram-quota-will-reach-118gb-in-2024-up-12-year-on-year

not possible to say how critical this challenge would be for adoption, and there are studies which indicate that the effects of choice overload, in general, may be exaggerated. Additionally, the challenge of choice overload can be tackled either by having a more standardized option alongside the existing choices or guiding users when they face multiple options. The standardized option, whichever it would be, should be good enough for most average use cases and pushed as the primary option for anyone who is interested in using local LLMs. For those who want more from LLMs such as further customization and manual configurations, other options can be considered. Main players of the consumer computing field, such as Microsoft and Apple, may have important roles in providing more standardized local AI solutions, including LLMs.

Despite being the contrast, local LLMs should not undermine the importance of online LLMs. Online counterparts are still good solutions for many use cases, and they should not be painted in a negative light, nor is this the point of this thesis either. In fact, the massive contributions made to the AI industry by innovations such as OpenAI's ChatGPT are notable and respectable in the big picture. Creating negative division between online and local implementations of LLMs would be counterproductive for the AI industry. That said, online LLMs are not to be seen as a one-fit-all solution and the public should still be aware of the choices, depending on what they wish to achieve. The user should still maintain control over their information and make decisions as to whether they want the data processed on their own device or in the cloud. The importance of this control can be attributed to a variety of factors ranging from personal preferences to requirements, such as existing data protection laws, which may prohibit cloud processing of certain types of data. Once users are aware of the options and alternatives, they can make informed decisions on a case-by-case basis on how they want to interact with the AI solutions. There is a strong case to be made that such informed decisions and control will contribute positively to the overall public perception of AI.

It is reasonable to expect that the existing trust issues for AI, including local implementations, will continue to persist. Many of the companies that do local AI development are, undeniably, big tech companies. For example, Google has Gemma series, Meta has Llama series and Microsoft has its own AI ecosystem (Copilot+) with NPU integration. Trust from consumers, however, can be very difficult to regain after it is lost. This is the case regardless of the reasons why the trust was initially lost. Addressing such challenges is difficult because consumer attitudes are hard to change. It is also important to remember that trust is subjective and while one person may not trust some implementation, another person may trust it entirely. This observation itself means that a one-fits-all answer to this issue does not exist.

In this thesis work, the most common, notable advantages for running local LLMs were covered. These advantages did not include knowledge, since in most cases it is not possible for the users of local LLMs, with limited hardware configurations, to achieve similar capabilities in this regard as online counterparts (such as ChatGPT) would have. Instead, the most meaningful advantages lean more towards the privacy-related factors. Enhanced privacy can be extremely beneficial for many users in different environments. On one hand, the user may be required to protect data by law. On the other hand, seeking enhanced privacy may be a voluntary choice due to skepticism towards external factors such as cloud service providers. Privacy aspect combined with the notion that even smaller models are capable of handling datasets, such as Excel sheets (*as seen in 2.2.3*), it seems rather apparent that local LLMs can offer significant value to many user groups, which is relevant in the context of the third research question, alongside the completed user study, which aims to observe the knowledge capabilities of a local LLM for the user-defined questions.

The user study in the paper creates confidence in the idea that regular computer users can present the kinds of questions to the local LLM that challenge its trustworthiness. In this regard, it was still important to receive questions from various angles from people of different backgrounds to effectively analyze some of the model's strengths and weaknesses more comprehensively. Additionally, it was important that fully independent test users present the questions, which effectively excludes potential room for bias by those who have studied the model, as none of the participants in the study had previous experience with Meta's Llama 3. Meta Llama3-8B was chosen for the study because it is more of a general LLM, which aims to support a broader range of different fields and use cases. Thus, in general, the 8-billion parameter model should perform reasonably well for general questions of different fields and be suitable for a regular home PC (standard 16GB RAM setting).

The purpose of the user study was to examine if it is possible to discover recognizable strengths and weaknesses of the model tailored for the target group of general users that took part in the study. As a result of the study, it was possible to analyze certain responses that were rated as wrong and try to discover repeating patterns in this regard, while still considering the existing black box issue of AI, which is also prevalent in local LLMs. Although different people may present more *difficult* questions for LLMs than others, thus challenging them more, this type of user feedback is still generally important for LLM studies. For example, just like the online counterparts of local LLMs would have a rating option for the given answers, models which are meant to be used locally can also be rated but may require more extensive user studies. Since the local LLMs are not connected to the internet, it is possible that feedback forms do not exist for some of the less common models. Therefore, the users of the locally installed models cannot rate the answers in such a way that the feedback given would directly reach the developer end. To respond to such challenges, these user studies may turn out helpful both during the creation of the models and afterwards for future improvements.

The results of the user study strongly indicate that local LLMs, specifically the 8-billion parameter variant used for the testing, can offer value to the users. Test participants reported 3 out of 4 (75%) of responses as either mainly or fully true once neutral midpoints were excluded. At 75% however, it is debatable whether the value ought to be considered significant or not. On the contrary, given that home-run local LLMs have certain shortcomings that are inevitable due to the hardware limitations, the results demonstrate that proper training (prioritization) can still lead to satisfactory outcomes for the users – that is, using the limited available space as efficiently as possible to serve the common, general use cases. Additionally, there are many relevant tasks that local LLMs are capable of, such as data handling and analysis, which were not covered by the user study. As a conclusion, it is strongly up to the end-user and their specific tasks and needs whether local LLMs can offer them significant value. Further research on different tasks, such as programming or data analysis, may be warranted to explore further capabilities of various locally installed LLMs for different user groups.

5 Conclusion

Local LLMs can offer value for the end-user. However, it is debatable whether the value can be considered significant in all instances, as the overall value or lack thereof heavily depends on the priorities and expectations of the said users. Since the advantages of local LLMs can be listed from the most significant such as data security, privacy and regulatory factors to the least significant such as potential speed improvements within a margin of a few seconds, personal preferences and priorities of the users should be highlighted when measuring the overall benefits that local LLMs can generally provide. It is likewise important to take the tradeoffs into account local LLMs may have in comparison to their online counterparts and whether the limited knowledge of size-limited local LLMs is sufficient to complete the requested tasks efficiently.

The user study of the thesis focused on general-use local LLMs, such as Meta Llama3-8B, and provided holistic inspection into the trend of utilizing local LLMs more broadly. The real-life use cases vary, and further research may be warranted to explore the capabilities and potentials of tailored local LLMs for more specific use cases such as programming, data analysis and handling of various datasets. Progress in hardware capabilities and accessible clients with proper documentation are essential for making local LLMs more accessible for various user groups and the most significant challenges hindering the usage are related to memory requirements and lack of language support, which appears to be a wider, *potentially* inevitable side effect with smaller size-limited models. Further research is likewise warranted to seek methods to mitigate the challenges outlined in this paper.

References

Aggarwal, D. (2025). Porting AnythingLLM to Windows on Snapdragon. Qualcomm. https://www.qualcomm.com/developer/blog/2025/01/porting-anythingllm-npu-windows-on-snapdragon

Alzahrani, N., Alyahya, H. A., Alnumay, Y., Alrashed, S., Alsubaie, S., Almushaykeh, Y., ... & Khan, H. (2024). When benchmarks are targets: Revealing the sensitivity of large language model leaderboards. arXiv preprint arXiv:2402.01781.

Awan, M. T., & Khan, K. (2022). Linux vs Windows: A Comparison of Two Widely Used Platforms. Journal of Computer Science and Technology Studies, 4(1), 41-53.

Baek, T. (2025). open-webui: User-friendly AI Interface. GitHub. https://github.com/open-webui/open-webui

Batt, S. (2024). Why 16GB of RAM Is the New Standard for Windows PCs. MUO. https://www.makeuseof.com/16gb-ram-new-standard-windows/

Chang, T. A., Arnett, C., Tu, Z., & Bergen, B. K. (2023). When is multilinguality a curse? language modeling for 250 high-and low-resource languages. arXiv preprint arXiv:2311.09205.

Chen, C., Feng, X., Li, Y., Lyu, L., Zhou, J., Zheng, X., & Yin, J. (2024). Integration of large language models and federated learning. Patterns, 5(12).

Cunningham, A. (2024). Microsoft is reworking Recall after researchers point out its security problems. Ars Technica. https://arstechnica.com/gadgets/2024/06/microsoft-makes-recall-feature-off-by-default-after-security-and-privacy-backlash/

Element Labs Inc. (2025). LM Studio. https://lmstudio.ai/

Ganesan, A. V., Varadarajan, V., Lal, Y. K., Eijsbroek, V. C., Kjell, K., Kjell, O. N., ... & Flek, L. (2024). Explaining GPT-4's Schema of Depression Using Machine Behavior Analysis. arXiv preprint arXiv:2411.13800.

Hudson Rock (2023). 100,000 Hackers Exposed from Top Cybercrime Forums. https://www.hudsonrock.com/blog/100-000-hackers-exposed-from-top-cybercrime-forums

JGraph. (2025). diagrams.net, draw.io. https://www.diagrams.net/

Jin, R., Du, J., Huang, W., Liu, W., Luan, J., Wang, B., & Xiong, D. (2024, August). A comprehensive evaluation of quantization strategies for large language models. In Findings of the Association for Computational Linguistics ACL 2024 (pp. 12186-12215).

Joo, D., Hadidi, R., Feizi, S., & Asgari, B. (2024). Endor: Hardware-Friendly Sparse Format for Offloaded LLM Inference. arXiv preprint arXiv:2406.11674.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. Communications of the ACM, 60(6), 84-90.

Kshetri, N. (2025). Economics of Open Source Software and AI Models. *Computer*, *58*(1), 134-141.

LMSYS Org (2025). LMArena [web service]. https://lmarena.ai/

Manchanda, J., Boettcher, L., Westphalen, M., & Jasser, J. (2024). The Open Source Advantage in Large Language Models (LLMs). arXiv preprint arXiv:2412.12004.

McCarthy, J., Minsky, M. L., Rochester, N., & Shannon, C. E. (2006). A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955. AI magazine, 27(4), 12-12

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics, 5(4), 115-133.

Menlo Research. (2025). Jan. GitHub. https://github.com/menloresearch/jan

Microsoft. (2025). Copilot [Large language model]. https://copilot.microsoft.com

Microsoft (2025). Shop Copilot+ PCs. Windows. https://www.microsoft.com/en-us/windows/copilot-plus-pcs

Mintplex Labs Inc. (2025). anything-llm. GitHub. https://github.com/Mintplex-Labs/anything-llm

Mistral AI (2024). Frontier AI in your hands. Mistral AI. https://mistral.ai/

Morales, J. (2024). Analysts say average laptop RAM quota will reach 11.8GB in 2024 — up 12% year-on-year. Tom's Hardware. https://www.tomshardware.com/laptops/analysts-say-average-laptop-ram-quota-will-reach-118gb-in-2024-up-12-year-on-year

Nielsen, J. (10, April). Usability heuristics for user interface design.

Nomic, Inc. (2025). nomic-ai/gpt4all. GitHub. https://github.com/nomic-ai/gpt4all

Ollama. (2025). Ollama. GitHub. https://github.com/ollama/ollama

OpenAI. (2025). ChatGPT (May 2024 version) [Large language model]. https://chat.openai.com

Regulation, P. (2016). Regulation (EU) 2016/679 of the European Parliament and of the Council. Regulation (eu), 679, 2016.

Rein, D., Hou, B. L., Stickland, A. C., Petty, J., Pang, R. Y., Dirani, J., ... & Bowman, S. R. (2024). Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Rohden, S. F., & Espartel, L. B. (2024). Consumer reactions to technology in retail: choice uncertainty and reduced perceived control in decisions assisted by recommendation agents. *Electronic Commerce Research*, *24*(2), 901-923.

Scheibehenne, B., Greifeneder, R., & Todd, P. M. (2010). Can there ever be too many options? A meta-analytic review of choice overload. Journal of consumer research, 37(3), 409-425.

Singh, S., Nan, Y., Wang, A., D'Souza, D., Kapoor, S., Üstün, A., ... & Hooker, S. (2025). The leaderboard illusion. arXiv preprint arXiv:2504.20879.

Team, G., Mesnard, T., Hardin, C., Dadashi, R., Bhupatiraju, S., Pathak, S., ... & Kenealy, K. (2024). Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Thomas Woodside and Helen Toner: Evaluating Large Language Models. CSET | Center for Security and Emerging Technology. July 17, 2024. https://cset.georgetown.edu/article/evaluating-large-language-models/

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M. A., Lacroix, T., ... & Lample, G. (2023). Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Turing, A. M. (1951). Can digital computers think?. The Turing Test: Verbal Behavior as the Hallmark of Intelligence, 111-116.

United Nations General Assembly. (1948). The universal declaration of human rights (UDHR).

Widder, D. G., Whittaker, M., & West, S. M. (2024). Why ‘open’AI systems are actually closed, and why this matters. Nature, 635(8040), 827-833.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., ... & Brew, J. (2020). Huggingface’s transformers: State-of-the-art natural language processing. arXiv 2019. *arXiv preprint arXiv:1910.03771*, *10*.

Zhao, J., Plaza-del-Arco, F. M., & Cercas Curry, A. (2024). Language Model Council: Benchmarking Foundation Models on Highly Subjective Tasks by Consensus. arXiv e-prints, arXiv-2406.