ChatGPT

# Open-Source vs. Closed-Source Generative AI Models (LLMs)

Large Language Models (LLMs) are powerful AI systems that generate text and code based on patterns learned from vast datasets. One key difference in today's AI landscape is **how these models are released and used**: some are **open-source** while others are **closed-source**. Understanding this distinction is crucial for IT students because it affects how we can **learn from, build with, and deploy** these AI models. Below, we provide a lecture-style overview of open vs. closed LLMs – including definitions, examples, pros/cons, implications, and ways to work with each.

## Open-Source LLMs: Definition and Characteristics

**Open-source LLMs** are models whose code and often **model weights** (the learned parameters) are made publicly available under permissive licenses. This means anyone can **inspect, use, modify, and distribute** the model, within the terms of its license. In other words, an open LLM is **"fully available, modifiable, and can be self-hosted"** [1]. Developers have access to the model's architecture and parameters, enabling them to run the model on their own hardware, fine-tune it with new data, or improve its code. The philosophy here is transparency and collaboration: these models thrive on community contributions and shared knowledge.

Some key characteristics of open-source LLMs include:

- *Transparency:* The internal workings (architecture, training data, etc.) are visible for examination. This builds trust because **accessible source code allows thorough audits for security and ethics issues** [2]. Anyone can evaluate how the model was trained and identify potential biases or errors.
- *Community-driven innovation:* Open models benefit from a global community of researchers and developers. **Anyone can contribute improvements or fixes**, accelerating innovation through collective effort [3]. This collaborative model often leads to rapid advancements, with enthusiasts fine-tuning models for specific tasks and sharing their results.
- *Flexibility and control:* Because you can modify the model, open LLMs offer **flexibility to customize** the behavior or specialize it for a niche domain. You also have **total control over how and where the model is used**, including keeping it on private infrastructure so data never leaves your environment [4].
- *Cost-effectiveness:* Open-source models are frequently free to obtain and use. Aside from computing costs to run them, there are typically **no API fees or licensing costs** [5]. This makes advanced AI accessible to individuals or organizations with limited budgets.
- *Example:* **Meta's LLaMA 2** (2023) is a prominent open-source LLM release. Meta provided the model weights (7B–70B parameters) openly to researchers and practitioners, allowing anyone to run and fine-tune LLaMA 2 for their needs. Many other open models exist (see Examples section below).

However, open-source LLMs also come with **responsibilities**. Using them requires technical know-how – you must manage the infrastructure, updates, and ensure security yourself. We'll discuss these challenges in detail later.

## Closed-Source LLMs: Definition and Characteristics

**Closed-source LLMs** are **proprietary models** developed by companies or organizations that **do not release the model's weights or full code to the public**. The model is essentially a black box – users can only interact with it through restricted channels (like an API or a web interface), and cannot see or alter its inner workings. In short, a closed LLM is **"commercial, provided behind an API, and controlled by a company"** [1]. The company maintains exclusive ownership of the model's architecture, training data, and parameters.

Characteristics of closed-source LLMs include:

- *Proprietary technology:* These models often represent the cutting edge of AI research, developed with massive resources. For example, OpenAI's **GPT-4** is a closed model that achieved state-of-the-art performance, but OpenAI **did not release technical details like model size, architecture, or training data** in its public report [6]. The exact "secret sauce" is kept confidential for competitive and safety reasons.
- *Access via services:* To use a closed LLM, you typically rely on the provider's service. This usually means **calling the model through a cloud API or a hosted platform** rather than running it on your own machine [7]. You send input (prompts) to the provider's servers and receive the model's output in return.
- *Controlled updates:* The maintaining company improves and updates the model over time, but users don't control these updates. You might notice the model getting better (or changing behavior) as the company pushes new versions behind the scenes.
- *Licensing and usage terms:* Using closed models comes with terms of service or licenses. There may be fees per request or monthly subscriptions, and restrictions on what you can do with the model (for instance, disallowing certain content or uses). Businesses often sign agreements to use these models in products, which include legal protections for the provider [8].
- *Examples:* **OpenAI's GPT-4** (accessible through ChatGPT or an API) is a prime example of a closed LLM. Other examples are **Google's PaLM 2** (which powers Google Bard), **Anthropic's Claude**, and **Cohere's Command models** – none of which provide their model weights publicly [9]. Users can query these models online, but cannot download or modify them directly.

Closed-source models prioritize polish, performance, and reliability, offering "AI as a service." They suit users who want out-of-the-box AI capabilities without delving into model internals. Next, we'll look at concrete examples of each type of model.

## Examples of Open-Source LLMs

Open-source LLM projects are flourishing, providing alternatives to proprietary models. Here are a few notable examples:

- **Meta LLaMA and LLaMA 2:** LLaMA (Large Language Model Meta AI) was introduced by Meta (Facebook) in early 2023, initially sharing weights with researchers. It sparked huge interest when its weights leaked publicly, leading to many fine-tuned variants. Meta then released **LLaMA 2** with an open license for everyone in mid-2023, in 7B, 13B, and 70B parameter versions. LLaMA 2 is **one of the most popular open LLMs** and can often approach the performance of older closed models like GPT-3 [10]. Its release allowed developers worldwide to run a high-quality model on their own hardware and build chatbots similar to ChatGPT openly.
- **BLOOM:** BLOOM is a 176-billion-parameter multilingual language model developed by the BigScience research collaboration (hundreds of researchers globally) and released in 2022 under

an open license. It was a milestone as an open model of similar scale to Google's largest models. BLOOM demonstrates how academic and open communities can produce powerful models; it's freely available for download and use [11].

- **GPT-NeoX and GPT-J:** These are open-source LLMs created by EleutherAI (an open research collective). GPT-J (6B parameters) and GPT-NeoX-20B (20B parameters) were attempts to replicate the capabilities of GPT-3 using open data and code. They are smaller than GPT-3, but being open, they allowed developers to host their own GPT-style model. They can be found on repositories like Hugging Face and have been integrated into various applications [11].
- **Falcon:** Falcon (40B and 7B versions) is an open-source LLM released by the Technology Innovation Institute (TII) in the UAE. It topped some performance leaderboards for open models in 2023. Falcon's weights are downloadable under the Apache 2.0 license, making it attractive for commercial use as well.
- **Vicuna and other derivatives:** Vicuna is not a base model but a **fine-tuned** open model worth mentioning. It was created by fine-tuning Meta's LLaMA on user-shared ChatGPT conversations. The result was a high-quality chatbot that, according to its creators, achieved ~90% of ChatGPT's quality at the time [12]. Vicuna exemplifies how open models can be improved by the community. Many such derivative models exist (Alpaca, WizardLM, etc.), building on base open models with specialized training.

These examples show the range of open LLM efforts, from large-scale projects backed by corporations (Meta's LLaMA 2) to community-driven endeavors (EleutherAI models, Vicuna by academics). All are united by a commitment to openness that lets users run and build upon the models directly.

## Examples of Closed-Source LLMs

On the closed-source side, the AI industry's most famous and capable models are proprietary. Here are key examples of closed LLMs:

- **OpenAI GPT-3.5 and GPT-4:** OpenAI's GPT series is what kicked off the current AI boom. **GPT-3.5** powers the initial version of ChatGPT, and **GPT-4** (released 2023) is OpenAI's latest flagship model known for its advanced reasoning and accuracy. GPT-4 is **accessible only via OpenAI's services** (e.g., ChatGPT Plus or the OpenAI API) – its architecture and training data are closely guarded secrets [6]. GPT-4 is multimodal (accepts images and text) and achieved top-level performance on many academic and professional benchmarks. Developers can use GPT-3.5/4 through an API, but cannot run these models locally. OpenAI continually refines these models (for example, improving their ability to follow user instructions or reducing biases) and pushes updates behind the scenes.
- **Google PaLM 2 / Bard and Gemini:** Google's **PaLM 2** is a family of advanced language models that power Google's Bard chatbot and other Google AI features. PaLM 2 is closed-source – Google has not released the weights, only some research publications. It excels in multilingual understanding and even coding. **Bard** (an AI chat service) provides an interface to PaLM 2 for end-users. Google has also announced **Gemini**, an upcoming multimodal model, which is expected to be another closed model accessible via Google Cloud services [13]. These models are part of Google's proprietary AI offerings, integrated into their products (Google Docs, Search, etc.) and available via API on Google Cloud, but not downloadable.
- **Anthropic Claude: Claude** is a large language model developed by Anthropic, an AI safety-focused startup. It's closed-source and provided via an API and a chat interface. Anthropic's approach emphasizes making Claude helpful, honest, and harmless, using techniques like "Constitutional AI" (a method for aligning the model's behavior with a set of principles) [14]. Claude 2, released in 2023, is comparable to ChatGPT in capability and sometimes gives more

detailed, less terse responses. Because Claude is proprietary, developers must get access from Anthropic (they offer a waitlisted API and beta web interface).
- **Others:** There are several other closed LLM providers. **Cohere** offers models like Command and Embed, accessible via API for enterprise NLP tasks. **AI21 Labs** has Jurassic-2, another large model accessible as a service. **IBM's Watsonx** is building LLM services. Even **Meta's own earlier chatbots (like Galactica)** were released as demos but not with open weights, making them effectively closed. Each of these is only usable through the company's tools and comes with usage agreements.

In summary, closed LLMs are typically developed by tech giants or well-funded startups and represent the frontier of capability. Users get access through APIs/online services, paying for the compute time in the cloud. Next, we'll compare the advantages and disadvantages of each approach.

## Advantages of Open-Source LLMs

Open-source LLMs offer a number of **benefits to developers, researchers, and organizations**:

- **Customization and Flexibility:** With open models, you can tailor the AI to your needs. You have access to the model's weights and code, so you can **fine-tune the model on domain-specific data or modify its behavior** directly [15]. This is great for specialized applications – for example, fine-tuning an open LLM for legal document analysis or medical report summarization. HatchWorks (a tech firm) notes that open-source LLMs give companies **flexibility to customize and integrate the model without waiting for a vendor's updates** [16].
- **Transparency and Trust:** Open models enable **full transparency**. Developers and auditors can inspect the architecture and even the training dataset (when provided) to understand how the model works. This openness builds trust, as issues can be spotted and addressed by the community. For instance, accessible code and weights allow **thorough security audits and bias checks** to ensure the model meets ethical standards [2]. In sensitive or regulated industries, this ability to **audit the AI** is a major advantage.
- **Community Collaboration:** A vibrant community often surrounds popular open-source models. This leads to shared **improvements, extensions, and support** from many contributors. For example, researchers worldwide contributed to making Vicuna (an open chatbot) by refining an existing open model. **Community support channels (forums, GitHub, etc.) are available** for open projects [3]. The collective intelligence of the community means bugs get fixed and new features added faster than one company might achieve alone. This crowdsourced innovation is a hallmark of open-source.
- **Lower Cost:** Open-source LLMs can significantly **reduce costs**. There are usually no licensing fees – once you download the model, it's yours to use. Compared to paying for API calls on a closed model, running an open model locally or on your own server can be cheaper at scale. An example: GPT-4's API usage was around $0.03 per 1K tokens, whereas an equivalently strong open model might run on infrastructure at a fraction of that cost [17]. Many open models are even free of charge entirely, allowing small startups or students to experiment without a big budget [5].
- **Data Privacy and Control:** Using an open LLM, you can deploy it **on-premises or on a private cloud**, so **your data never leaves your controlled environment**. For anyone concerned about sensitive data (e.g., hospitals with patient data, or companies with proprietary info), this is crucial. With an open model, you can comply with strict privacy requirements by keeping all inputs/outputs in-house. **You're not sending data to a third-party service** to get AI results [18], which mitigates risks of leaks or breaches on an external server.

In summary, open-source LLMs **democratize AI** – they put powerful tools in the hands of many. You can tinker, learn from them, and innovate on top of them. They foster an environment of **shared progress**, much like open-source software has done in general.

## Challenges of Open-Source LLMs

Despite their benefits, open-source LLMs come with several **challenges or disadvantages** that students and developers should be aware of:

- **Technical Complexity:** Running a state-of-the-art LLM isn't trivial. Large models demand **powerful hardware (GPUs or TPUs), lots of memory, and engineering expertise** to deploy efficiently [19]. Even smaller open models can be tricky to get running if you're not experienced with machine learning frameworks. Fine-tuning a model also requires knowledge of ML training procedures. In short, open models are **DIY** — you must handle installation, scaling, and optimization yourself, which can be a barrier if you lack resources.
- **Maintenance and Support Burden:** Unlike paid services, open-source projects usually **don't come with guaranteed support or SLAs** (Service Level Agreements). If something goes wrong, you rely on community forums or your own team to fix it. For enterprises, this can be a drawback – there's no dedicated helpdesk if the model crashes. As one source points out, open LLM users do not get **dedicated support teams or robust service guarantees** as a vendor would provide [20]. This means adopting an open model requires confidence in your in-house capabilities to maintain it.
- **Security & Compliance Responsibility:** With a closed service, a company might handle security patches and compliance certifications for you – but with an open model, **the onus is on the user to operate it securely and meet regulations** [21]. If you deploy an open LLM in a product, you need to implement measures to protect data and prevent misuse. There's also a risk that openly available models could be used by malicious actors (since they're freely accessible), which raises concerns about releasing powerful models without oversight. Organizations must be vigilant in how they use and secure open AI tools.
- **Resource and Funding Limitations:** Many open-source LLM projects are community or academia-driven, which often means **limited computing resources and funding** compared to tech giants. This can lead to slower development cycles and less robust testing. As one article notes, open models often rely on volunteers, so there may be **fewer resources for fixing bugs, adding features, and optimizing performance** [22]. In practice, some open LLMs might not be as finely tuned or stable as commercial offerings that undergo rigorous QA.
- **Variable Quality and Performance:** Not every open model will match the top closed models in capability. Quality can vary widely – some open LLMs are excellent (approaching GPT-3/4 level on certain tasks), while others are very specialized or experimental. A Charter Global report highlighted that **not all open models perform at the level of top closed-source alternatives – some are more narrow in scope** [23]. Users must carefully choose an open model that suits their needs, and sometimes accept a trade-off in raw performance or polish. Additionally, open models might lack the extensive fine-tuning for safety that commercial models undergo, so they may be more prone to problematic outputs unless users apply their own filters.

In short, using open-source LLMs means **taking on more responsibility**. You get freedom and control, but you also inherit all the work of running and improving the model. This is often worth it for the flexibility – but it's important to plan for the engineering effort required.

## Advantages of Closed-Source LLMs

Closed-source, proprietary LLMs remain very attractive in many scenarios due to the following advantages:

- **State-of-the-Art Performance:** The companies behind closed models invest heavily in research, data, and compute to push model quality to the maximum. These models often **lead the field in capabilities and accuracy**, since they can leverage massive training runs on private datasets. For example, OpenAI's GPT-4 and Google's latest models generally outperform most open models on complex reasoning, coding, and knowledge benchmarks. Proprietary models like GPT-4 are **"backed by extensive R&D investments, leading to state-of-the-art performance."** [24] In short, if you need the very best model available for a task, it's often a closed model at the cutting edge.
- **Easy Integration (Plug-and-Play):** Using a closed-source model is typically as simple as calling an API – no need to set up servers or manage libraries. This **fast, plug-and-play access** means you can get results immediately and scale usage without worrying about deployment details [25]. The providers handle the heavy lifting of hosting the model on cloud infrastructure. For developers, this convenience can dramatically speed up development. For instance, an app developer can send text to the OpenAI API and get a GPT-4 answer in one line of code, instead of dealing with GPU setups. Closed models are **optimized for high reliability and low latency in production environments** [26], which is great for enterprise use.
- **Managed Maintenance and Updates:** With closed models, all the **maintenance, updates, and improvements are handled by the provider**. OpenAI, Google, Anthropic, etc., continuously tune and upgrade their models (often based on user feedback or new research) and roll out these updates to users automatically. This means the model you use via API might quietly get better over time without you changing anything. Users "don't need to worry about model updates, hardware, or scaling – the provider handles all of that" [27]. This is a huge relief in a production setting, as you outsource the engineering effort to experts. It also often includes **uptime guarantees** and robust infrastructure: you can rely on the service to be available and performant when you need it.
- **Security and Compliance Features:** Leading AI providers implement strong security measures and obtain certifications for their cloud services. For businesses in regulated industries, using a closed-source LLM through a vendor can help with compliance, because **vendors often invest in security, privacy controls, and compliance with regulations like GDPR or HIPAA** [28]. They may offer data encryption, data residency options, audit logs, and other enterprise features. For example, Microsoft's Azure OpenAI Service (which hosts GPT-4) can certify compliance and offer private network connections. This can make closed models suitable for sensitive applications where an open DIY solution might face hurdles to meet all security requirements.
- **Dedicated Support and Documentation:** Commercial model providers usually offer **extensive documentation, developer tools, and customer support**. For instance, OpenAI provides detailed API docs, usage guidelines, and has a support team for enterprise customers. Google Cloud has support plans for its AI services. This means if you hit a problem or need help integrating the model, you have official channels to turn to. Such **high-level support – integration assistance, troubleshooting, etc. – is a benefit of closed models** [29]. In a mission-critical setting, having vendor support can be reassuring. The documentation and best-practice guides that come with closed APIs also help developers get up to speed quickly.

Overall, closed-source LLMs excel in providing a **polished, ready-to-use AI experience**. They minimize the effort on the user's side and maximize performance out-of-the-box. This is why many companies start with closed models to prototype and scale AI solutions rapidly.

## Challenges of Closed-Source LLMs

On the flip side, relying on closed-source models has some notable downsides:

- **Cost and Licensing:** Closed LLMs typically **cost money to use**, and those costs can accumulate quickly. Providers charge per API call or per number of tokens processed. For heavy usage (say, an app that makes thousands of queries), this can become expensive. **Usage is usually billed per token or request, which can become costly at scale** [30]. Small startups or student projects might find the fees limiting, especially for the most advanced models. Additionally, there may be tiered subscription plans or upfront commitments for enterprise access. Beyond just money, you are also bound by the provider's license terms – violating them (e.g., using the model for disallowed purposes) could result in losing access.
- **Limited Customization:** When using a closed model, you **cannot alter the model's architecture or training data**. You more or less get a one-size-fits-all solution. This lack of flexibility means if the model's behavior isn't ideal for your task, you have limited recourse except to engineer clever prompts or wait for the provider to improve it. While some services allow slight customization (for example, OpenAI has offered fine-tuning for certain models, but with constraints), it's nowhere near the freedom of open models. As one source notes, users **"generally cannot access the model internals or fine-tune it beyond limited parameters," restricting domain-specific adaptations** [31]. For researchers, this is a severe limitation – you can't probe how the model works internally or modify it for experiments.
- **Transparency and Trust Issues:** Closed models are **black boxes**. The lack of insight into training data or algorithmic logic makes it hard to **identify biases, errors, or reasons behind outputs**. You have to trust the provider's claims about the model. This opacity can be problematic: for instance, if a closed model produces a biased result, you can't easily inspect why because the training corpus is unknown. A SpringsAI article points out that with closed models, **users cannot fully understand how decisions are made or address potential biases, due to the lack of transparency** [32]. In fields where explainability is important (law, healthcare), this could be a serious drawback of closed LLMs.
- **Vendor Lock-In and Dependency:** Using a closed-source API means you become **dependent on the provider**. If they change pricing, throttle the service, experience outages, or even shut down a model, your application is directly affected. This risk is often termed *vendor lock-in*. Companies might find it hard to switch away if they've built a lot around a specific model's API. Also, if the provider's business fails or they decide to discontinue the model, you're left stranded. As Charter Global notes, relying on third-party APIs carries **risks like service outages, API changes, or price hikes** [33]. You are at the mercy of the provider's business decisions and stability.
- **Data Privacy Concerns:** With closed models, you usually must send your data (prompts) to an external server (the model host). This raises concerns if the data is sensitive. Even if the provider claims to not store or use your input data beyond serving your request, some organizations are uncomfortable with any external transmission. **Your data is essentially in a third party's hands during processing** [18]. In response, some providers offer on-premise installations for a premium, but in general, using a closed API means relinquishing a degree of control over data handling. For students and developers, this might limit using closed models with confidential datasets unless proper agreements are in place.

In summary, closed-source LLMs trade away control and openness for convenience and performance. Users must **trust the provider** and accept constraints, which can be perfectly fine for many uses but problematic for others. The lack of transparency and potential for lock-in are key considerations when choosing a closed solution.

## Implications for Development and Research

The open vs. closed model dichotomy has important consequences for how developers and researchers can **advance AI or build new applications**:

For **research and learning**, open-source LLMs are invaluable. They allow students and scientists to **peer inside the model**, experiment with different training methods, and reproduce results. Openness promotes academic rigor – findings can be verified because the same model can be examined by others. Indeed, open models tend to **"promote reproducibility and adhere to ethical research standards"** under public scrutiny [34]. Researchers can probe an open model's biases or limitations and publish improvements, pushing the field forward. For example, much of the knowledge on how to fine-tune and interpret language models comes from experiments on open architectures like GPT-2 or open-sourced portions of models. Conversely, closed models limit research: it's hard to verify claims or build on results when the model is proprietary. A prominent case was **GPT-4's release**, which included a paper without revealing basic details like model size or training data. This drew criticism from the scientific community, as it **made independent verification and understanding nearly impossible** [35]. In short, if the future of AI were entirely closed, it could slow down academic progress. The presence of open models ensures that **researchers have platforms to innovate on and study**, which is essential for the scientific ecosystem.

For **developers and innovators**, the choice influences development style and speed. With open models, a developer can create a highly customized solution – they can embed the model in an application and tweak it for optimal results. This is ideal for use-cases that require fine control or on-premise deployment (for example, a medical app using an LLM that must run in a hospital's secure network). However, this comes at the cost of complexity: integrating an open model might mean setting up GPUs, optimizing inference speed, and dealing with model updates manually [36]. It demands a stronger ML engineering skillset on the team. On the other hand, closed models let developers **add AI features quickly using simple API calls**, without needing ML infrastructure. This lowers the barrier to entry – even those with little AI background can incorporate, say, GPT-4 into a project by following API docs. The trade-off is less flexibility: you might have to design your app around the constraints of the API (rate limits, input/output formats) and you can't fix the model's flaws yourself. In practice, many developers start prototyping with a closed API (for speed) and later, if scale or customization becomes an issue, consider an open-source model deployment.

In the big picture, many expect a **hybrid future** for development and research, where open and closed models co-exist. Closed models will continue driving the frontier of what's possible (since companies will pour resources into them), but open models will ensure that the broader community can participate and that knowledge disseminates. In fact, "most experts see a hybrid future — where companies and communities co-evolve models together" [37]. For students, this means it's wise to get familiar with both approaches: understand how to use APIs of closed models *and* how to fine-tune or deploy open models. Both skill sets will be valuable.

## Implications for Industry Use

When it comes to industry and business adoption, open vs. closed LLMs is a strategic decision. Companies must weigh **control vs. convenience, cost vs. performance, and risk vs. reward**:

*Enterprise survey results on adopting open-source LLMs.* In one 2024 survey, **41% of enterprises said they plan to increase their use of open-source models**, and another 41% would switch to open models if they achieve performance parity with closed models (only 18% foresaw no increase in open-source use)

[38]. This shows a strong interest in open-source LLMs across industry, primarily driven by the desire for **greater control, customizability, and cost savings** [39]. Many businesses have found that if an open model can deliver similar results, the benefits of owning the solution (without vendor lock-in or usage fees) are very attractive.

That said, as of 2023, closed-source models dominated enterprise AI deployments (an estimated 80–90% of LLM usage was through closed platforms) [40]. The reasons are understandable: closed models like GPT-4 provided best-in-class performance and came with enterprise support and compliance guarantees that organizations trust. For critical applications where accuracy and reliability are paramount, companies have been willing to pay for the top proprietary model. Furthermore, not all organizations have the AI expertise to self-host an open model at scale – many prefer a turnkey solution from a reputable vendor. **Security and compliance** also play a role: certain industries might choose a closed provider who offers certified compliance (e.g., SOC2, HIPAA) over taking on the liability themselves, even if an open model is technically free.

However, the trend is shifting. As open models improve in quality, more companies are piloting them. The survey results above suggest that **over 80% of enterprises are open to using more open-source LLMs, given sufficient performance** [38]. If open models continue to narrow the gap with closed models, we may see a more **even split in adoption (around 50/50)** in the coming years [40]. Organizations cite **control over the technology (no black-box dependence)** and **lower long-term costs** as key motivators for moving to open LLMs [39]. Being able to audit the model and ensure data never leaves their own systems is a huge plus, especially for finance, healthcare, and government sectors.

In practice, many industries might adopt a **hybrid approach**: using closed-source LLM services for some tasks and open-source models for others [41]. For instance, a company could use a closed API like GPT-4 for a customer-facing chatbot where accuracy is critical, but use an open model internally for processing sensitive internal documents to keep that data in-house. Another scenario is starting with a closed model to test an idea (quick deployment), then switching to an open model when scaling up to reduce costs.

For IT students, the takeaway is that **both open and closed models have roles in industry**. Understanding the business implications – like how open models can mitigate vendor risk but require in-house talent, or how closed models can speed up development but incur ongoing costs – will help you make informed decisions in your future projects or workplaces. The landscape is evolving as open-source communities become more formidable (sometimes matching what only big corporations could do before). We are likely heading toward an ecosystem where **companies leverage the strengths of each approach**: open-source for control and innovation, closed-source for the latest features and convenience, depending on the use case.

## Using and Contributing to Open vs. Closed Models

Finally, as students and developers, how can you **interact with these models** and even contribute to their development? The processes differ significantly between open and closed LLMs:

- **Working with Open-Source LLMs:** You can freely **download open models** and run them yourself. Websites like Hugging Face Hub host dozens of open LLMs (LLaMA 2, BLOOM, etc.) that you can obtain with a click or a `git` command [42]. With a compatible library (such as Hugging Face's Transformers in Python), you can load the model on your machine or a cloud instance and start generating text. Many open models are optimized to run on a single GPU or even CPU for smaller versions. As a student, you might use a smaller 7B or 13B model on a personal laptop with enough RAM or Google Colab. You can also **fine-tune** open models on custom data – for example, fine-tuning an open model on Shakespeare's works to mimic Shakespearean style – using frameworks like LoRA or full training if you have the compute. Contributing to open-source LLM projects is also possible: if you have improvements (better training data, new model architectures, or even just bug fixes in the code), you can contribute via platforms like GitHub. Some students contribute by joining large collaborations (like BigScience for BLOOM) or by creating derivative models and releasing them for others. The barrier to entry is low: even experimenting with prompts and sharing your findings on forums is a way to contribute to community knowledge.

- **Using Closed-Source LLMs (APIs and Platforms):** To use a closed model like GPT-4 or Claude, you typically **sign up with the provider and use their API or app**. For example, OpenAI offers a web interface (ChatGPT) and a developer API. To use the API, you would get an API key from OpenAI and install their SDK (e.g., `openai` Python package), then call a function in your code to send a prompt and receive the model's completion [43]. It's straightforward: no need to manage the model's environment. As a student, you can take advantage of free tiers or academic credits that some providers offer (OpenAI has free trials, and Anthropic's Claude might have a limited free web demo, etc.). When integrating into applications, closed providers have **detailed documentation and examples** to follow [7]. However, remember you are bound by their usage terms – e.g., you shouldn't input truly sensitive personal data unless the terms allow and you trust the service's privacy. In terms of contributing, improving a closed model's **internals** isn't possible unless you work at the company. But you can still contribute indirectly: companies often welcome user feedback to correct model errors or biases. For instance, using ChatGPT and pressing the "thumbs down" on a bad answer with an explanation actually helps OpenAI refine future versions. Some closed-model companies also release research papers or open-source smaller components (OpenAI has open-sourced tools and smaller models in the past). As a developer, you can contribute by building **cool applications** on top of these APIs – demonstrating new use cases or identifying limitations. This kind of work can influence the direction of the technology (for example, the popularity of applications built on GPT-3's API in 2021 helped signal demand for more features, guiding development).

In summary, **to get hands-on experience**, try both approaches. You might download an open model (like a 7B parameter LLaMA 2) and run some prompts locally to see how it works. You could also sign up for a service like OpenAI or Cohere and practice calling their API. This will give you intuition about differences: e.g., an open model might require some effort to get running but can be tweaked, whereas an API call is instant but you cannot change the model's behavior except through the prompt. Both are useful skills. Importantly, always respect the licensing – open models might have restrictions (some allow commercial use, some don't), and closed models definitely have usage policies.

By engaging with open-source, you join a community of developers pushing AI forward collaboratively. By using closed-source services, you gain access to the latest and greatest models for your projects. Many developers use **a combination** – for example, prototyping an idea with a free open model, then later switching to a paid API for better output, or vice versa. Being versatile with both types will serve you well in the AI field.

## Conclusion

Both open-source and closed-source generative models have critical roles in the AI ecosystem, each with its own philosophy and trade-offs. **Open-source LLMs** embody the spirit of openness: they democratize access to AI, encourage learning and innovation, and give users control over the technology. **Closed-source LLMs**, in contrast, drive the cutting edge of performance and provide reliable, turnkey AI solutions, albeit under the stewardship of private companies. As we've discussed, there is no one "right" approach – the choice depends on context and needs. Developers and businesses often find **value in a hybrid approach**, leveraging the strengths of both open and closed models [37].

For you as IT undergraduates, this topic sets the stage for understanding how AI will evolve. If you value transparency, community, and the ability to tinker under the hood, you might lean into open-source projects – maybe even contribute to one in the future. If you're drawn to building high-impact applications quickly using the best available tools, you'll likely integrate proprietary models via APIs. Most importantly, **the AI revolution is not happening in secret labs alone; it's happening in the open as well**, and you can be a part of it. Whether you choose to fine-tune a model on your own data or plug into a world-class model through an API, the knowledge you've gained about both paradigms will help you navigate the exciting opportunities in the world of generative AI.

---


[1] [7] [9] [10] [11] [13] [15] [37] [42] [43] Open Source vs Closed Source LLMs: Everything You Need to Know (and How to Use Them!) | by Rohan Mistry | Medium
https://medium.com/@rohanmistry231/open-source-vs-closed-source-llms-everything-you-need-to-know-and-how-to-use-them-bec324d47ba6

[2] [3] [16] [17] [36] [38] [39] [40] Open-Source LLMs vs Closed: Unbiased Guide for Innovative Companies [2025]
https://hatchworks.com/blog/gen-ai/open-source-vs-closed-llms-guide/

[4] [5] [6] [8] [12] [14] [22] [29] [32] Open Source vs Closed Source LLMs. Pros and Cons - Springs
https://springsapps.com/knowledge/open-source-vs-closed-source-llms-pros-and-cons

[18] [19] [20] [21] [23] [25] [26] [27] [28] [30] [31] [33] [41] Open-Source vs Closed-Source LLM Software | Charter Global
https://www.charterglobal.com/open-source-vs-closed-source-llm-software-pros-and-cons/

[24] [34] [35] L'Atelier - L'Atelier
https://atelier.net/insights/evaluating-open-source-large-language-models