PDFMathTranslate: Scientific Document Translation Preserving Layouts

Rongxin Ouyang[1,†], **Chang Chu**[2,†,*], **Zhikuang Xin**[3], **Xiangyao Ma**[4]
[1]National University of Singapore, Singapore. [2]Tsinghua University, Beijing, China.
[3]University of Chinese Academy of Sciences, Beijing, China.
[4]Funstory.ai Limited, Hong Kong SAR, China.
[†] These authors contribute equally to this work.
[*]**Correspondence:** chuc23@mails.tsinghua.edu.cn.

Abstract

Language barriers in scientific documents hinder the diffusion and development of science and technologies. However, prior efforts in translating such documents largely overlooked the information in layouts. To bridge the gap, we introduce **PDFMathTranslate**, the world's first open-source software for translating scientific documents while preserving layouts. Leveraging the most recent advances in large language models and precise layout detection, we contribute to the community with key improvements in precision, flexibility, and efficiency. The work is open-sourced at https://github.com/byaidu/pdfmathtranslate with more than 222k downloads.

1 Introduction

Science and technologies diffuse and develop in different languages (Von Gizycki, 1973; Montgomery, 2013). Yet, language barriers hinder the diffusion and development of scientific progress (Ramírez-Castañeda, 2020; Hwang, 2005; Ammon, 2012). For example, while 98% publications in science are written in English (Liu; Ammon, 2012), the language only has 7.3% native speakers and no more than 20% speakers worldwide (Bahji et al., 2023). Consequently, the majority of human beings are hindered by language barriers from the advances in science and technologies. To overcome the barrier, there are the attempts from international organizations for inclusive science [1] and the attempts from academia in improving *Machine Translation (MT)* over text (Brown et al., 1990; Vaswani et al., 2017; Zhu et al., 2020; Johnson et al., 2017; Sennrich et al., 2016).

However, text-based machine translation fails to address the unique challenges posed by the layouts in *technical translation* (Schubert, 2012) where the *elaborate access structure* matters (pp. 351-352). Non-textual elements in scientific and technical documents are not ignorable — the arrangement of paragraphs, mathematical equations, tables, and figures have rich and important meanings. Thus, text-based machine translation ignoring such important information appear insufficient to address the barrier in scientific and technical document translation, hindering the progress of science, technologies, and society.

To fill the gap, we introduce **PDFMathTranslate**, the world's first open-source tool to translate PDF documents with preserved layouts. By leveraging recent advances in layout detection and large language models (see Figure 1), we better address the barrier with at least five key contributions: (1) efficient workflow of layout detection, translation, and re-rendering; (2) support for multiple languages; (3) support for multiple translation models and services; (4) diverse user interfaces; and (5) a community-commerce model affording sustainable developments.

2 Architecture and Design

PDFMathTranslate is designed to translate documents while preserving their original layouts. First, it accepts a PDF document along with user-specified parameters, such as languages and the preferred translation service. Next, it detects the layouts to extract the layouts and textual contents from the document. Third, the texts are translated using the selected translation service, such as GPT-4, DeepL, Google, or Ollama (see Appendix A). Finally, the translated text and previously detected layouts are re-rendered as a translated document with preserved layouts.

Technically, we design the architecture for *precision*, *flexibility*, and *efficiency*. These three principles are realized by a precise parser, a flexible translation middle-ware, and an efficient workflow, detailed in following subsections.

[1]See https://www.unesco.org/en/open-science

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 918–924

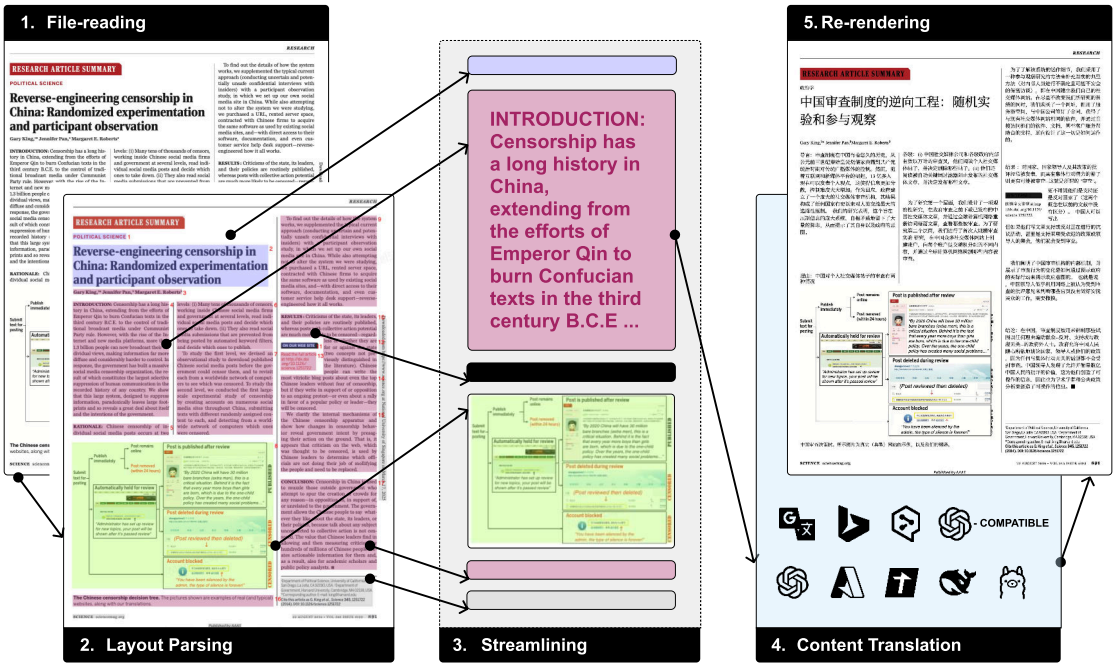

Figure 1: The architecture of PDFMathTranslate

2.1 Precise layout parser

Preserving layouts requires precise layout extraction. To precisely parse the position of document elements, we propose a pipeline consisting of layout detection, splitting, processing, and re-rendering. In the beginning, we exploit a recent advance in layout detection, *DocLayout-YOLO-DocStructBench-onnx*. The model is another version of a SOTA solution in object detection, *Yolov10* (Wang et al., 2024), and is fast and accurate in the specific task (Zhao et al., 2024).

To increase the compatibility of this model, we support two model formats: the *Open Neural Network Exchange (ONNX)* [2] standard and a *pytorch* version. The ONNX version was chosen as default to ensure the compatibility of our parsing pipeline for diverse hardware.

2.2 Flexible translation middleware

Translation middleware offers flexibility in terms of *language diversity*, *supported services*, and *customization capabilities*. One, PDFMathTranslate supports at least 194 languages (see Appendix A), including popular ones such as *English*, and some languages shared by relatively smaller communities, such as *Cherokee*. Users can freely select any of these languages as either the input or output language when translating documents.

Two, PDFMathTranslate supports at least 23 popular services for translation. The list of services includes prominent online services such as Google Translate and OpenAI, as well as local deployments of models for either translation or dialogs (see Appendix A). In addition to explicitly listed services, users can integrate any new service complying with OpenAI protocols [3].

Moreover, to simplify the addition of new services, we designed a structure that separates layout flows from translation flows. In this structure, language services receive only textual inputs, enhancing the generalizability and sustainability of our project. As a result, implementing any language service requires only a concise function of fewer than 15 lines.

Three, PDFMathTranslate provides additional customizability in translation. We designed a customization feature prompting strategies that potentially increase the quality of translation or its adaptability in domain-specific tasks. This feature allows users to exploit advanced prompting strategies such as few-shots (Brown et al., 2020), Chain-of-Thought (Wei et al., 2022), Role-playing (Kong

[2] See `https://onnx.ai/`

[3] See: https://platform.openai.com/docs/api-reference/chat/create

et al., 2024; Shanahan et al., 2023), etc.

2.3 Efficient streaming flow

Translating scientific and technical documents often necessitates an efficient workflow, especially when dealing with thousands of pages, which can challenge overall performance. To address the issue, we landed on a streaming design that is centered around an efficient in-memory processing pipeline that minimizes disk I/O while preserving document fidelity. Specifically, the `translate_stream` function accepts a byte stream of a PDF file and transforms it into a mutable document representation. This approach allows the system to dynamically apply modifications, such as converting to PDF/A format when needed and embedding the appropriate fonts based on the target language. By employing temporary buffers and leveraging functions like `download_remote_fonts`, the design effectively manages resources and ensures that the original file remains unaltered during processing.

Our workflow supports various translation backends, with translators configured as plugins within our workflow. Different translators integrate distinct translation functionalities by inheriting from a base class and implementing the `do_translate` method. These translators can be seamlessly integrated into different user interfaces. With contributions from the community, our workflow currently supports more than 20 translators. Additionally, it provides robust extensibility for backend LLM services with the OpenAI-compatible API and traditional machine translation service.

Further optimization in performance is achieved through a combination of asynchronous execution, caching strategies, and error-handling mechanisms. Asynchronous constructs, such as `asyncio.Event`, are employed to support task cancellation and concurrency, particularly when processing large documents or handling multiple translation tasks simultaneously. In addition, the design incorporates a caching mechanism—controlled via the `ignore_cache` flag—to prevent redundant computations, save LLM tokens, and accelerate the translation process. While a subset font embedding strategy minimizes the final file size without compromising compatibility. Overall, this streaming design reflects a robust, scalable, and resource-efficient framework tailored for automated PDF translation workflows.

Figure 2: Officially supported interfaces

Figure 3: Major commands for CLI

3 Usage and Deployments

PDFMathTranslate has been implemented in various interfaces, including a command-line tool (CLI), a graphic user interface (GUI), cross-platform applications (on MacOS and Windows), and Docker images. In addition to those officially supported interfaces, user `guaguastandup` contributes a Zotero plugin to our community.

For laymen, we offer GUI on Mac [4], Windows, Web, online demos, and Zotero plugin [5]. Specifically, desktop versions on Mac and Windows can be installed, demos are publicly accessible online, and GUI can be started using `pdf2zh -i`. Those graphic interfaces are shown in Figure 2.

For developers, we support CLI and Docker for advanced usage and substream developments. Developers who prefer command-line tools can conveniently install our Python CLI program using `pip install pdf2zh`. The command-line tool supports extensive features, with the major options shown in Figure 3 and more documented in details [6]. For users who want to deploy the software on servers, we offer a Docker image for simplified deployment. Additionally, Docker images are built and distributed on different platforms for developers with potential network issues.

```
docker pull byaidu/pdf2zh
```

[4] See https://github.com/reycn/pdf2zh-mac.

[5] See https://github.com/guaguastandup/zotero-pdf2zh; community-contributed.

[6] See https://github.com/Byaidu/PDFMathTranslate/blob/main/docs/ADVANCED.md

```
docker run -d -p 7860:7860 byaidu/
    pdf2zh
```

4 Comparison with Existing Systems

Compared with existing alternatives, PDFMathTranslate offers a superior solution from three perspectives: *accessibility*, *readability*, and *efficiency* (see Table 1).

In terms of *accessibility*, the tool is open-sourced, self-deployable, supports API calls, and is entirely free of charge. These advantages ensure wider accessibility compared to any other existing solutions. Regarding *readability*, the tool preserves layouts, supports complex formulas, and is capable of exporting bilingual documents. Notably, a current limitation of our tool is its lack of optimal optical character recognition (OCR) support, which restricts its wider application for scanned documents. However, OCR functionality is planned for implementation in upcoming versions. Finally, concerning *efficiency*, while this application is slower than text-based translation services (like Google Translate) due to the integration of precise layout detection models, our solution is significantly faster than alternative products that also preserve layouts (such as IMT and Doc2X).

5 Use Cases

We illustrate two typical use cases of PDFMathTranslate to illustrate how the tool translates text while preserving information embedded in layouts. The first use case (Panel A, Figure 4) is an excerpt from a textbook with both text and complex formulas, and the second case (Panel B, Figure 4) is a scientific research article with complex layouts and figures (see Figure 4). Both cases consistently show that our work is capable of precisely translating the information within text while preserving the crucial information embedded in layouts.

6 Sustainability

Nevertheless, maintenance is challenging to open-source projects (Stol and Ali Babar, 2010). To ensure sustainable development, we've established a *community-commerce* model, through which the incentives of developers are provided by two sources: open-source recognition and the benefits of commercial products.

First, we increase developer exposure by prominently featuring contributors on the project's homepage and regularly highlighting recent contributions (e.g., *1970-01-01 / Supports Google translation (by @author_handle)*). This strategy has successfully incentivized 44 global developers who collectively contributed tens of thousands of lines of code.

Second, we provide sponsored rewards such as membership exchanges for active contributors through collaboration with a commercial partner. These 11 subscriptions sent out to developers have effectively encouraged consistent contributions, resulting in the resolution of over 485 user-reported issues (by April 2025). Such cooperation has enabled the integration of advanced academic and community-driven technologies, including ONNX model support (Wybxc, 2025) and Qwen (ws051682, 2025), thus advancing cross-linguistic scientific communication.

This sustainable collaboration model propelled the project to the top of GitHub's global trends for over a week, garnering more than 25k stars, 222k downloads, and over 49k Docker pulls (by June 2025). Additionally, our project's success has inspired a range of new commercial products with similar functionalities, significantly impacting both the scientific community and the broader public.

7 Limitations

While we have made several key improvements, our tool still has specific limitations. First, translation quality remains highly dependent on the underlying translation models and prompts. Second, the accuracy of the layout detection model in identifying various document layouts similarly depends on the quality of the layout detection models employed. Third, the current version of the tool can not handle scanned PDFs lacking optimal optical character recognition (OCR); however, this last feature is planned for implementation.

8 Acknowledgments

The authors thank all 44 contributors[7] of the open-sourced project on GitHub; platforms allowing us

[7] They are 7shi, Byaidu, Copilot, Cycloctane, Hanaasagi, IuvenisSapiens, JEFF-dev-ui, Tql-ws1, Wybxc, YadominJinta, Zxis233, alohays, aseaday, awwaawwa, borcation, charles7668, chiu0602, czz404, damaoooo, dependabot[bot], domonnss, eltociear, hellofinch, highkay, hotwa, imClumsyPanda, kharkover, kidach1, lintian233, mydreamworldpolly, namazuchin, qqueing, reycn, tastelikefeet, timelic, treeleaves30760, tylzh97, ws051682, wx-11, xxnuo, xyzyx233, yidasanqian, ymattw, zqqian; ordered alphabetically, retrieved through the official GitHub API by March 2025)

		OURS	IMT[5]	Doc2X	TeX based[6]	Google	DeepL
Accessibility	Open-source	✓			✓		
	Deployment	✓			✓		
	API integration	✓		✓		✓	✓
	Price	**Free**	Paid	Paid	Free	Free	Paid
Readability	Layout[1]	✓	✓	✓			
	Formula[2]	✓	✓	✓	✓		
	Bilingual[3]	✓	Partial	Partial	✓	✓	✓
	OCR[4]	✓[7]	✓	✓		✓	✓
Efficiency	Batch tasks	✓		✓	✓	✓	✓
	Speed (sec/page)	**1.47**	1.50	1.86	1.67	0.38	1.88

[1] Whether the tool can preserve figures, images, and formulas (although the formulas may be incorrectly displayed).
[2] Whether the tool can translate documents containing formulas and correctly handle the positions of the formulas.
[3] Whether the tool supports bilingual exports where both the original and translated text are simultaneously readable.
[4] Whether the tool can handle scanned documents instead of digital documents.
[5] IMT: Immersive Translate PDF Pro.
[6] The original LaTeX file is required for translation.
[7] Available in a community fork: `PDFMathTranslate/PDFMathTranslate-next`.

Table 1: Comparison with other projects and productions

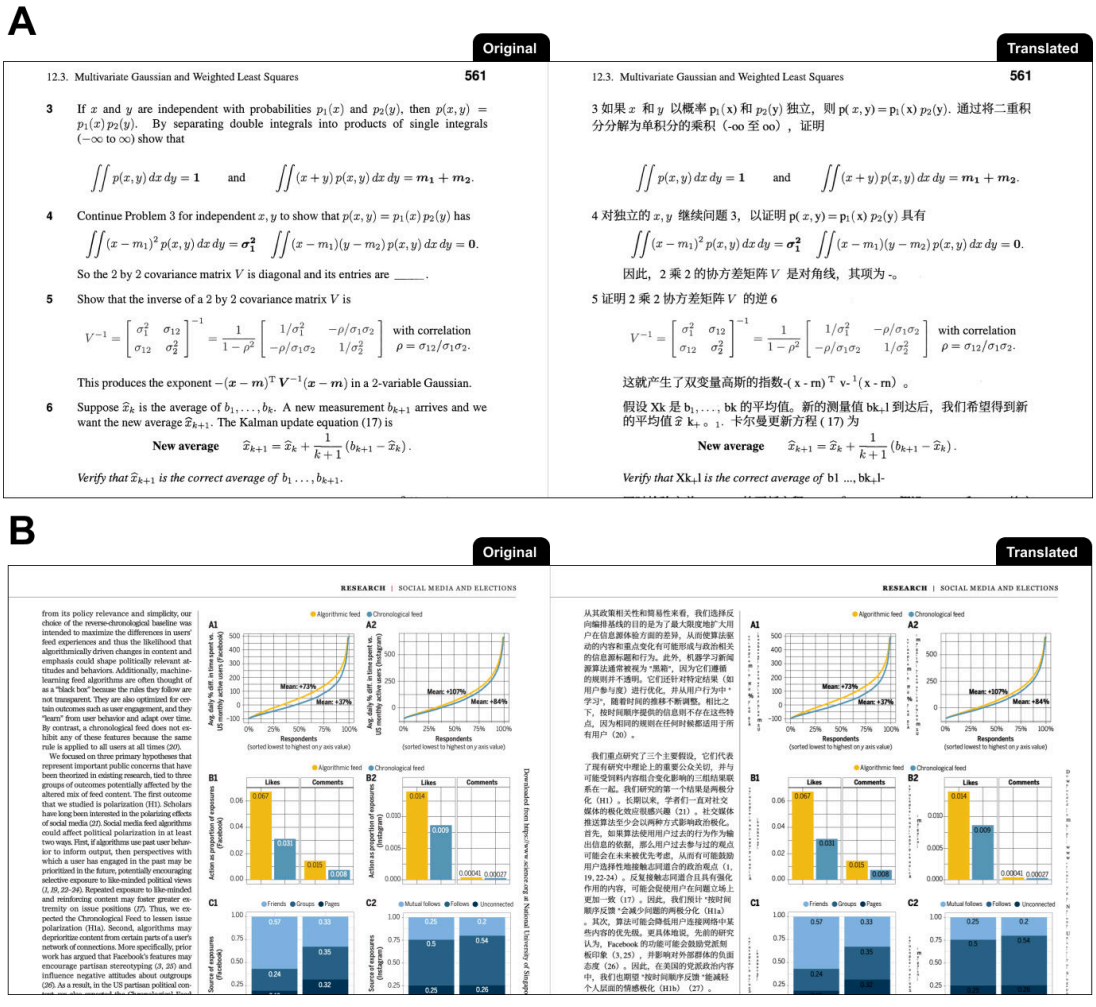

Figure 4: Use cases.

to host demos (Hugging Face, ModelScope); services we used in the demo (e.g., Google Translate); redeem codes from Immersive Translate; public API access from SiliconFlow; and all comments from users.

9 Ethics and Broader Impacts

Our work has made a significant positive impact within both multilingual scientific and open-source communities. However, as a flexible, widely used tool, it raises potential ethical concerns regarding the copyright of documents. Translating documents without proper permissions could challenge the intellectual property rights of scientific works and innovations. We welcome suggestions from experts in intellectual property to mitigate the concern.

References

Ulrich Ammon. 2012. Linguistic inequality and its effects on participation in scientific discourse and on global knowledge accumulation – With a closer look at the problems of the second-rank language communities. *Applied Linguistics Review*, 3(2):333–355.

Anees Bahji, Laura Acion, Anne-Marie Laslett, and Bryon Adinoff. 2023. Exclusion of the non-English-speaking world from the scientific literature: Recommendations for change for addiction journals and publishers. 40(1):6–13.

Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. 1990. A Statistical Approach to Machine Translation. 16(2):79–85.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.

Kumju Hwang. 2005. The inferior science and the dominant use of english in knowledge production: A case study of korean science and technology. *Science Communication*, 26(4):390–427.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, and Greg Corrado. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. 5:339–351.

Aobo Kong, Shiwan Zhao, Hao Chen, Qicheng Li, Yong Qin, Ruiqi Sun, Xin Zhou, Enzhi Wang, and Xiaohang Dong. 2024. Better zero-shot reasoning with role-play prompting. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4099–4113.

Weishu Liu. The changing role of non-English papers in scholarly communication: Evidence from Web of Science's three journal citation indexes. 30(2):115–123.

Scott L. Montgomery. 2013. *Does Science Need a Global Language?: English and the Future of Research*. University of Chicago Press.

Valeria Ramírez-Castañeda. 2020. Disadvantages in preparing and publishing scientific papers caused by the dominance of the english language in science: The case of colombian researchers in biological sciences. *PloS one*, 15(9):e0238372.

Klaus Schubert. 2012. Technical translation. In *Handbook of Translation Studies: Volume 1*, pages 350–355. John Benjamins Publishing Company.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. *Preprint*, arXiv:1508.07909.

Murray Shanahan, Kyle McDonell, and Laria Reynolds. 2023. Role play with large language models. 623(7987):493–498.

Klaas-Jan Stol and Muhammad Ali Babar. 2010. Challenges in using open source software in product development: A review of the literature. In *Proceedings of the 3rd International Workshop on Emerging Trends in Free/Libre/Open Source Software Research and Development*, pages 17–22. ACM.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

Rainald Von Gizycki. 1973. Centre and Periphery in the International Scientific Community: Germany, France and Great Britain in the 19th Century. 11(4):474–494.

Ao Wang, Hui Chen, Lihao Liu, Kai Chen, Zijia Lin, Jungong Han, and Guiguang Ding. 2024. Yolov10: Real-time end-to-end object detection. *arXiv preprint arXiv:2405.14458*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. 35:24824–24837.

ws051682. 2025. feat(translator): Add Ali Qwen MT translator by ws051682 · Pull Request #585 · Byaidu/PDFMathTranslate. [Online; accessed 18. Mar. 2025].

Wybxc. 2025. feat: onnx support by Wybxc · Pull Request #116 · Byaidu/PDFMathTranslate. [Online; accessed 18. Mar. 2025].

Zhiyuan Zhao, Hengrui Kang, Bin Wang, and Conghui He. 2024. DocLayout-YOLO: Enhancing Document Layout Analysis through Diverse Synthetic Data and Global-to-Local Adaptive Perception. *Preprint*, arXiv:2410.12628.

Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tie-Yan Liu. 2020. Incorporating BERT into Neural Machine Translation. *Preprint*, arXiv:2002.06823.

A Appendix

A.1 Officially supported language and services

In Table 2, we enumerate the officially supported services, models, and languages. However, the actual number may be significantly higher due to the extensibility of services and models.

Table 2: Supported languages and services for translation in PDFMathTranslate

Category	Details	Total
Service or models	Google *(default)*, 302.AI, Bing, DeepL, DeepLX, Ollama, Ali Qwen, Ollama, Xinference, Gemma, OpenAI, OpenAI-like , DeepSeek, AzureOpenAI, Zhipu, ModelScope, Silicon Cloud, Gemini, Azure, Tencent, Dify, AnythingLLM, Argos Translate, Grok, Groq	*23*
Input and output languages	Abkhaz, Acehnese, Acholi, Afrikaans, Albanian, Alur, Amharic, Arabic, Armenian, Assamese, Awadhi, Aymara, Azerbaijani, Balinese, Bambara, Bashkir, Basque, Batak Karo, Batak Simalungun, Batak Toba, Belarusian, Bemba, Bengali, Betawi, Bhojpuri, Bikol, Bosnian, Breton, Bulgarian, Buryat, Cantonese, Catalan, Cebuano, Chichewa (Nyanja), Chinese (Simplified), Chinese (Traditional), Chuvash, Corsican, Crimean Tatar, Croatian, Czech, Danish, Dinka, Divehi, Dogri, Dombe, Dutch, Dzongkha, English, Esperanto, Estonian, Ewe, Fijian, Filipino (Tagalog), Finnish, French, French (French), French (Canadian), Frisian, Fulfulde, Ga, Galician, Ganda (Luganda), Georgian, German, Greek, Guarani, Gujarati, Haitian Creole, Hakha Chin, Hausa, Hawaiian, Hebrew, Hiligaynon, Hindi, Hmong, Hungarian, Hunsrik, Icelandic, Igbo, Iloko, Indonesian, Irish, Italian, Japanese, Javanese, Kannada, Kapampangan, Kazakh, Khmer, Kiga, Kinyarwanda, Kituba, Konkani, Korean, Krio, Kurdish (Kurmanji), Kurdish (Sorani), Kyrgyz, Lao, Latgalian, Latin, etc.	*194*

[1] The number of supported languages far exceeds what is shown in the table. Our protocol, designed as a general framework for OpenAI-like services, allows users to integrate any translation system, including locally hosted models or private online services, ensuring extensive language support.
[2] Based on the default service (Google Translate) and interface (CLI).