The Journal of Open Source Software

SLMat: A Comprehensive Serverless Toolkit for Advanced Materials Design

Kamal Choudhary [1]

1 DeepMaterials LLC, Silver Spring, MD, USA

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Summary

SLMat is a serverless, browser-based toolkit that offers a scalable and efficient alternative to traditional server-based platforms. By eliminating the need for server management and providing persistent setups, SLMat enhances productivity and security, enabling researchers to focus on innovation rather than infrastructure. The toolkit integrates seamlessly with materials databases, supports AI model development, and offers advanced features like AI-guided coding and chatbot integration. SLMat accelerates production time, promotes reproducibility, and democratizes access to computational resources, making it a versatile solution for researchers across various domains.

Statement of Need

The rapid evolution of computational materials science is marked by significant advances in high-throughput simulations, machine learning (ML) models, and the extensive use of large-scale databases (Schleder et al., 2019). As researchers continue to push the boundaries of materials discovery and design, the need for robust, flexible, and scalable computational tools becomes increasingly apparent. Server-based platforms such as Google Colab (Bisong & Bisong, 2019) have become integral to this research landscape, offering powerful environments for running tutorials, complex simulations, and computational workflows. However, these platforms present challenges, including server resource management, potential latency issues, re-installations of packages for each session, and privacy concerns when handling sensitive data such as security tokens. SLMat addresses these challenges by offering a serverless materials design toolkit that is accessible directly from a web browser (Baldini et al., 2017). SLMat is based on platforms such as Pyodide and JupyterLite (Droettboom, 2019; Randles et al., 2017).

Progressive Web Application (PWA)

SLMat is built as a Progressive Web Application (PWA) (Biørn-Hansen et al., 2017), designed to provide a seamless and reliable user experience across various platforms, including desktops, laptops, and mobile devices. The core architecture of SLMat decouples the graphical user interface (GUI) from the computational backend, allowing for computations to be performed either locally in the browser or remotely through distributed cloud services. This design eliminates the need for server management, offering a simplified backend that reduces operational complexity and cost.

One of the significant advantages of SLMat is its in-browser computing capability. By enabling computations to be conducted directly within the web browser, SLMat removes the dependency on external servers, which enhances security by keeping data local and provides a more responsive and faster user experience. This serverless approach ensures that computational resources can be dynamically scaled according to the user's needs, making SLMat suitable for a wide range of applications, from quantum simulations to AI model development.

Complementing Existing Server-Based Platforms

While server-based platforms like Google Colab have been instrumental in advancing materials science research, they often require users to manage server resources and navigate the complexities of cloud computing environments. These platforms can introduce latency, particularly when reinstalling packages for each session, processing large datasets, or running resource-intensive simulations, and may raise privacy concerns when sensitive data is processed on remote servers.

SLMat addresses these issues by providing a complementary serverless option. This toolkit allows researchers to perform advanced computational tasks without worrying about server management or data privacy issues. The serverless nature of SLMat ensures that users can leverage local computational resources while still benefiting from the scalability and flexibility typically associated with cloud computing. This approach reduces operational costs and accelerates production time by streamlining the computational workflow.

Integration with Materials Design Tools and Databases

SLMat is designed to integrate seamlessly with existing materials design tools and databases, enhancing its utility for a broad range of research applications. By supporting popular materials databases (Choudhary et al., 2020; Jain et al., 2013), SLMat enables researchers to access and analyze extensive datasets directly within the platform. This integration facilitates the combination of data from multiple sources, allowing for comprehensive analyses and the identification of new materials with optimized properties.

In addition to database integration, SLMat supports the use of advanced materials design tools (Choudhary & Garrity, 2024; Larsen et al., 2017), enabling researchers to perform simulations, molecular dynamics studies, and other computational experiments with ease. The platform's flexibility allows it to be used in conjunction with other tools, such as scikit-learn (Pedregosa et al., 2011), ensuring that researchers can incorporate these resources into their workflows without the need for additional configuration or setup.

AI Model Development and Chatbot Integration

SLMat supports AI surrogate model development with packages such as Scikit-learn and XGBoost (Chen & Guestrin, 2016). By incorporating both commercial and open-access AI-guided coding tools such as ChatGPT (Liu et al., 2023) and HuggingFace (Wolf et al., 2019), SLMat helps researchers write code more efficiently and with greater accuracy, even if they are not experts in programming. These AI-driven features provide real-time suggestions and code completions based on the context of the user's work, significantly reducing the time and effort required to develop complex models.

SLMat also offers the unique capability of chatbot integration, providing researchers with a conversational interface to assist with various tasks. This integration allows users to interact with the platform using natural language, making it easier to navigate the toolkit's features, retrieve information, and even run simulations. The chatbot can be used to automate repetitive tasks, answer common questions, and guide users through the process of setting up and executing computational workflows, further streamlining the research process.

Security and Cost Efficiency

In the current landscape of computational research, security and cost efficiency are critical considerations (Shafiei et al., 2022). SLMat addresses these concerns by offering a serverless architecture that enhances security. Since computations can be performed locally within the browser, sensitive data remains on the user's device, reducing the risk of exposure associated with cloud-based platforms. This security advantage makes SLMat particularly suitable for research areas where data confidentiality is paramount, such as in biomedical materials science.

Moreover, the simplified backend of SLMat reduces the overhead associated with server management, leading to lower operational costs. By eliminating the need for dedicated servers and minimizing the complexity of the computational environment, SLMat provides a cost-effective solution for researchers, allowing them to focus on their scientific inquiries rather than the logistics of resource management.

Improved Production Time and Reproducibility

The streamlined workflow offered by SLMat contributes to improved production time, enabling researchers to move from concept to execution more quickly. The platform's intuitive interface, combined with its AI-guided coding and chatbot features, reduces the time required to set up and run simulations, allowing for faster iteration and experimentation.

Furthermore, SLMat promotes reproducible research (Choudhary et al., 2024; Peng, 2011) by providing a consistent and standardized environment for computational materials design. The ability to share workflows and results within the platform ensures that other researchers can replicate studies and build upon existing work, fostering collaboration and accelerating the pace of discovery in materials science.

In the future, SLMat's capabilities can be further enhanced by integrating it with in-house computing clusters or cloud providers (Wankhede et al., 2020), offering a hybrid approach that combines the flexibility of serverless architecture with the power of dedicated computational resources. This integration would allow researchers to seamlessly scale their workloads, leveraging local clusters for high-performance tasks while utilizing cloud resources for additional scalability and storage as needed.

Figures

This schematic highlights SLMat's key features: serverless architecture with in-browser computation, simplified backend, lower costs, and enhanced security. It integrates multiscale materials design tools, connects to major materials databases, supports AI model development with persistent setups, and includes chatbot integration for an intuitive user experience.

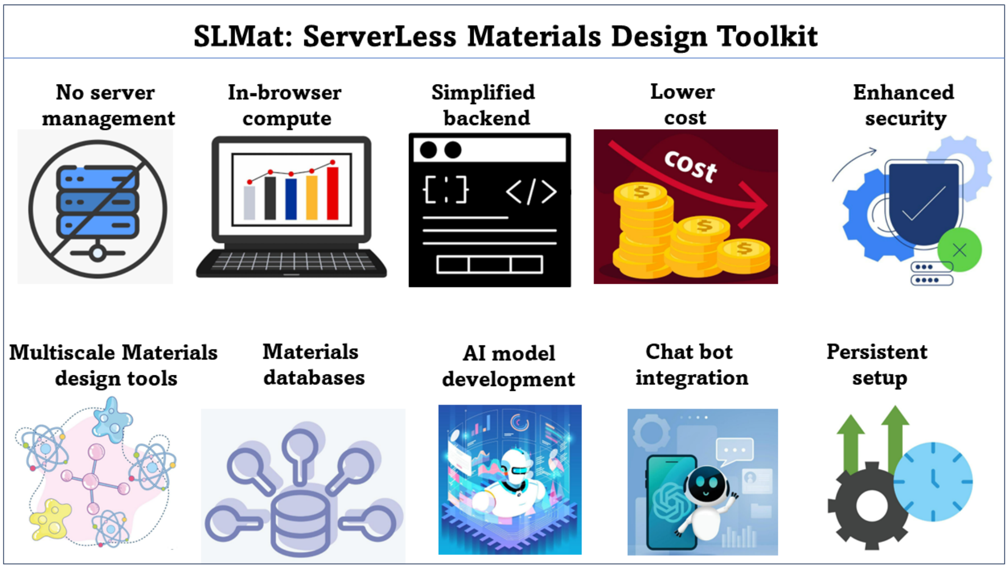

Figure 1: SLMat schematic

References

1. Schleder, G. R., Padilha, A. C., Acosta, C. M., Costa, M., & Fazzio, A. (2019). From DFT to machine learning: Recent approaches to materials science–a review. *Journal of Physics: Materials*, 2(3), 032001.
2. Bisong, E. (2019). Google Colaboratory. In *Building Machine Learning and Deep Learning Models on Google Cloud Platform* (pp. 59-64). Apress, Berkeley, CA.
3. Baldini, I., Castro, P., Chang, K., Cheng, P., Fink, S., Ishakian, V., … & Suter, P. (2017). Serverless computing: Current trends and open problems. In *Research Advances in Cloud Computing* (pp. 1-20). Springer, Singapore.
4. Droettboom, M. (2019). Pyodide: Bringing the scientific Python stack to the browser-Mozilla Hacks-the Web Developer Blog. Mozilla Hacks.
5. Randles, B. M., Pasquetto, I. V., Golshan, M. S., & Borgman, C. L. (2017). Using the Jupyter notebook as a tool for open science: An empirical study. In *2017 ACM/IEEE Joint Conference on Digital Libraries (JCDL)* (pp. 1-2). IEEE.
6. Biørn-Hansen, A., Majchrzak, T. A., & Grønli, T.-M. (2017). Progressive web apps: The possible web-native unifier for mobile development. In *International Conference on Web Information Systems and Technologies* (Vol. 2, pp. 344-351). SciTePress.
7. Choudhary, K., Garrity, K. F., Reid, A. C., DeCost, B., Biacchi, A. J., Hight Walker, A. R., … & Tavazza, F. (2020). The joint automated repository for various integrated simulations (JARVIS) for data-driven materials design. *npj Computational Materials*, 6(1), 1-13.
8. Jain, A., Ong, S. P., Hautier, G., Chen, W., Richards, W. D., Dacek, S., … & Persson, K. A. (2013). Commentary: The Materials Project: A materials genome approach to accelerating materials innovation. *APL Materials*, 1(1), 011002.
9. Larsen, A. H., Mortensen, J. J., Blomqvist, J., Castelli, I. E., Christensen, R., Dulak, M., … & Jacobsen, K. W. (2017). The atomic simulation environment—a Python library for working with atoms. *Journal of Physics: Condensed Matter*, 29(27), 273002.
10. Choudhary, K., & Garrity, K. F. (2024). InterMat: Accelerating band offset prediction in semiconductor interfaces with DFT and deep learning. *Digital Discovery*, 3(7), 1365-1377.
11. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine

Learning Research, 12, 2825-2830.

12. Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785-794).
13. Liu, Y., Han, T., Ma, S., Zhang, J., Yang, Y., Tian, J., … & Liu, Z. (2023). Summary of ChatGPT-related research and perspective towards the future of large language models. *Meta-Radiology*, 100017.
14. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., … & Rush, A. M. (2020). Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.
15. Shafiei, H., Khonsari, A., & Mousavi, P. (2022). Serverless computing: A survey of opportunities, challenges, and applications. *ACM Computing Surveys*, 54(11s), 1-32.
16. Peng, R. D. (2011). Reproducible research in computational science. *Science*, 334(6060), 1226-1227.
17. Choudhary, K., Wines, D., Li, K., Garrity, K. F., Gupta, V., Romero, A. H., … & Tavazza, F. (2024). JARVIS-Leaderboard: A large scale benchmark of materials design methods. *npj Computational Materials*, 10(1), 93.
18. Wankhede, P., Talati, M., & Chinchamalatpure, R. (2020). Comparative study of cloud platforms-Microsoft Azure, Google Cloud Platform and Amazon EC2. *J. Res. Eng. Appl. Sci.*, 5(02), 60-64.

Baldini, I., Castro, P., Chang, K., Cheng, P., Fink, S., Ishakian, V., Mitchell, N., Muthusamy, V., Rabbah, R., Slominski, A., & others. (2017). Serverless computing: Current trends and open problems. *Research Advances in Cloud Computing*, 1–20.

Biørn-Hansen, A., Majchrzak, T. A., & Grønli, T.-M. (2017). Progressive web apps: The possible web-native unifier for mobile development. *International Conference on Web Information Systems and Technologies*, *2*, 344–351.

Bisong, E., & Bisong, E. (2019). Google colaboratory. *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners*, 59–64.

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. *Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining*, 785–794.

Choudhary, K., & Garrity, K. F. (2024). InterMat: Accelerating band offset prediction in semiconductor interfaces with DFT and deep learning. *Digital Discovery*, *3*(7), 1365–1377.

Choudhary, K., Garrity, K. F., Reid, A. C., DeCost, B., Biacchi, A. J., Hight Walker, A. R., Trautt, Z., Hattrick-Simpers, J., Kusne, A. G., Centrone, A., & others. (2020). The joint automated repository for various integrated simulations (JARVIS) for data-driven materials design. *Npj Computational Materials*, *6*(1), 173.

Choudhary, K., Wines, D., Li, K., Garrity, K. F., Gupta, V., Romero, A. H., Krogel, J. T., Saritas, K., Fuhr, A., Ganesh, P., & others. (2024). JARVIS-leaderboard: A large scale benchmark of materials design methods. *Npj Computational Materials*, *10*(1), 93.

Droettboom, M. (2019). Pyodide: Bringing the scientific python stack to the browser-mozilla hacks-the web developer blog. *Mozilla Hacks–the Web Developer Blog*, *16*.

Jain, A., Ong, S. P., Hautier, G., Chen, W., Richards, W. D., Dacek, S., Cholia, S., Gunter, D., Skinner, D., Ceder, G., & others. (2013). Commentary: The materials project: A materials genome approach to accelerating materials innovation. *APL Materials*, *1*(1).

Larsen, A. H., Mortensen, J. J., Blomqvist, J., Castelli, I. E., Christensen, R., Dułak, M., Friis, J., Groves, M. N., Hammer, B., Hargus, C., & others. (2017). The atomic simulation environment—a python library for working with atoms. *Journal of Physics: Condensed Matter*, *29*(27), 273002.

Liu, Y., Han, T., Ma, S., Zhang, J., Yang, Y., Tian, J., He, H., Li, A., He, M., Liu, Z., & others. (2023). Summary of chatgpt-related research and perspective towards the future of large language models. *Meta-Radiology*, 100017.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., & others. (2011). Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, *12*, 2825–2830.

Peng, R. D. (2011). Reproducible research in computational science. *Science*, *334*(6060), 1226–1227.

Randles, B. M., Pasquetto, I. V., Golshan, M. S., & Borgman, C. L. (2017). Using the jupyter notebook as a tool for open science: An empirical study. *2017 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, 1–2.

Schleder, G. R., Padilha, A. C., Acosta, C. M., Costa, M., & Fazzio, A. (2019). From DFT to machine learning: Recent approaches to materials science–a review. *Journal of Physics: Materials*, *2*(3), 032001.

Shafiei, H., Khonsari, A., & Mousavi, P. (2022). Serverless computing: A survey of opportunities, challenges, and applications. *ACM Computing Surveys*, *54*(11s), 1–32.

Wankhede, P., Talati, M., & Chinchamalatpure, R. (2020). Comparative study of cloud platforms-microsoft azure, google cloud platform and amazon EC2. *J. Res. Eng. Appl. Sci*, *5*(02), 60–64.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., & others. (2019). Huggingface's transformers: State-of-the-art natural language processing. *arXiv Preprint arXiv:1910.03771*.