JUNE 03, 2025 | AI ENGINEER WORLD'S FAIR

Introduction to LLM serving with SGLang

YINENG ZHANG & PHILIP KIELY

baseten

Welcome to the workshop!

Thank you for choosing to spend the morning with us working on inference optimization.
Let's have a great time and make some fast models.

Yineng Zhang

Core maintainer of SGLang, member LMSYS Org, model performance engineer at Baseten. Contributor to 3 papers including FlashInfer. Previously held engineering roles at Baidu and Meituan.

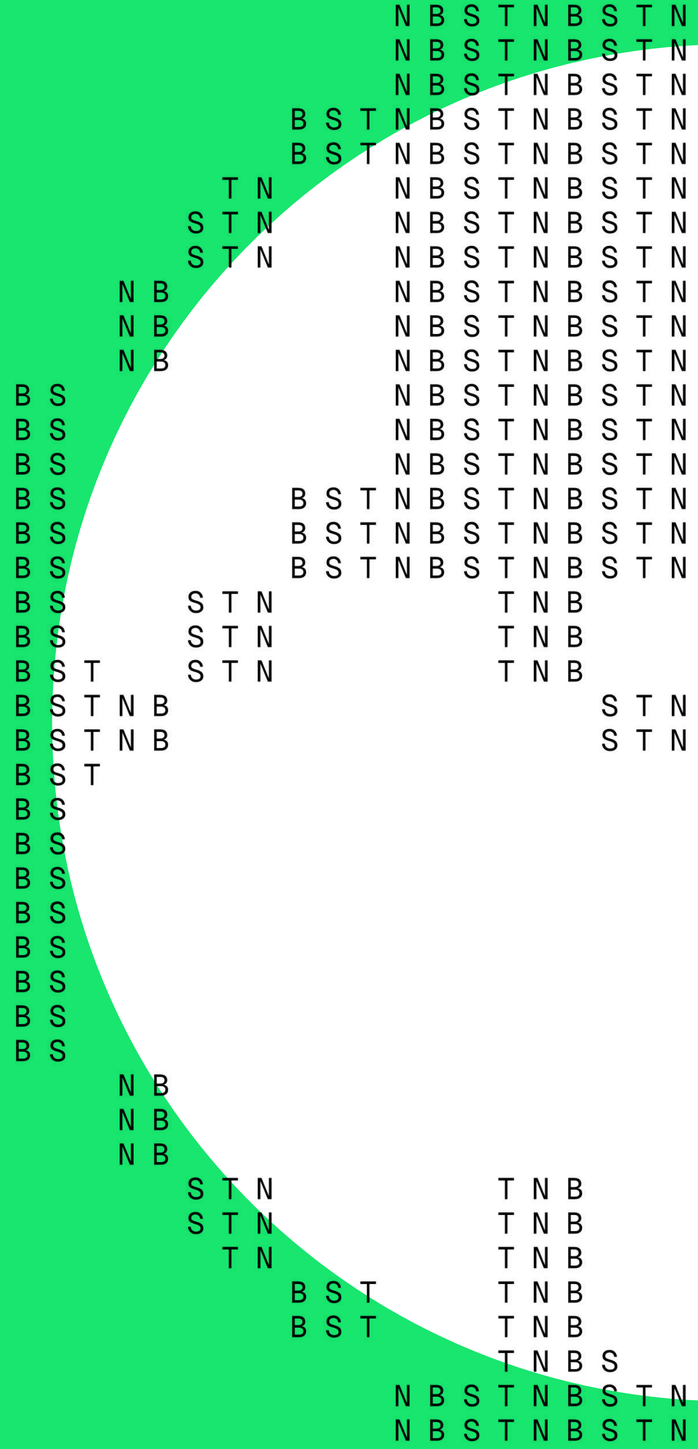
Philip Kiely

Head of developer relations at Baseten. B+ in linear algebra.

Agenda

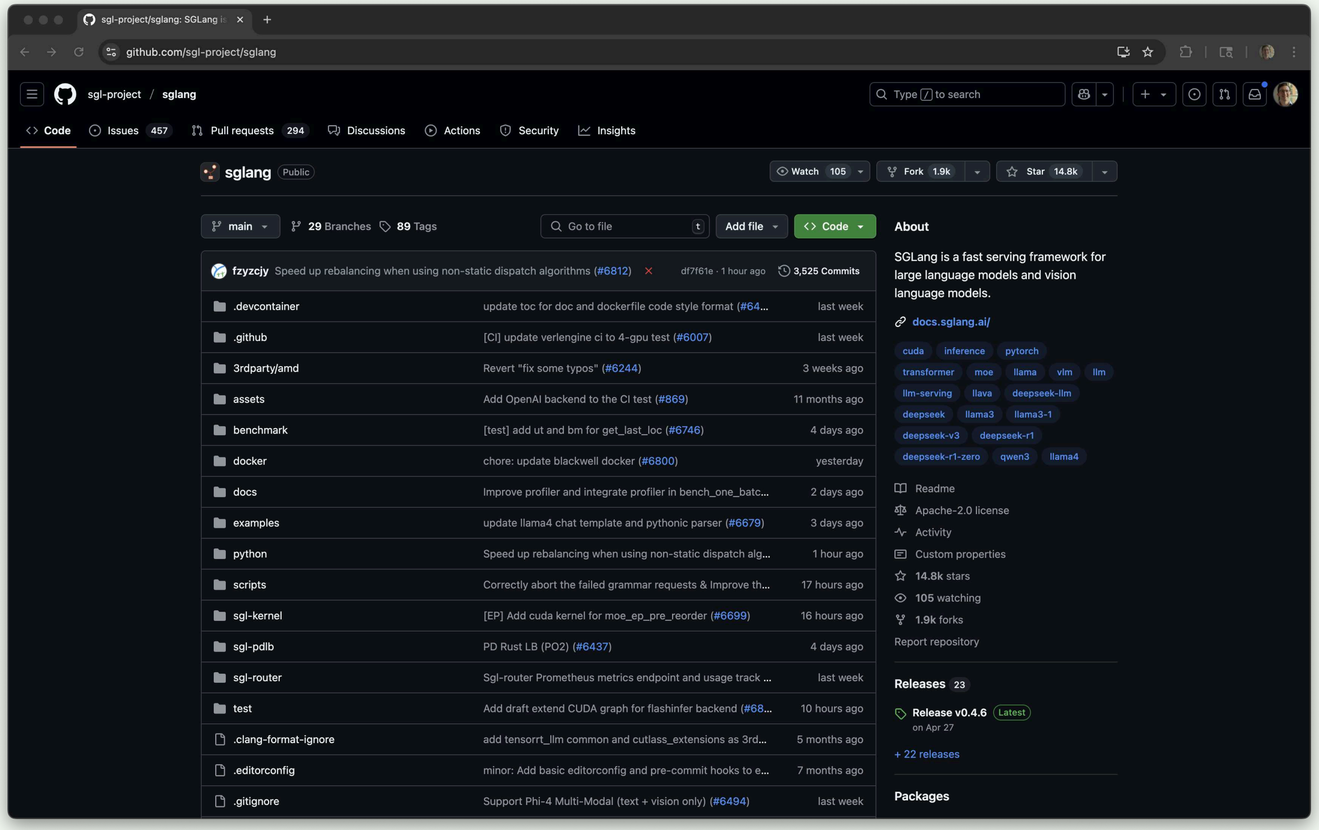
1. Introductions and setup
2. History of SGLang
3. Deploying your first model with SGLang
4. SGLang performance optimizations
5. SGLang community and codebase

Introduction



What is SGLang?

SGLang is an open-source fast serving framework for large language models and vision language models



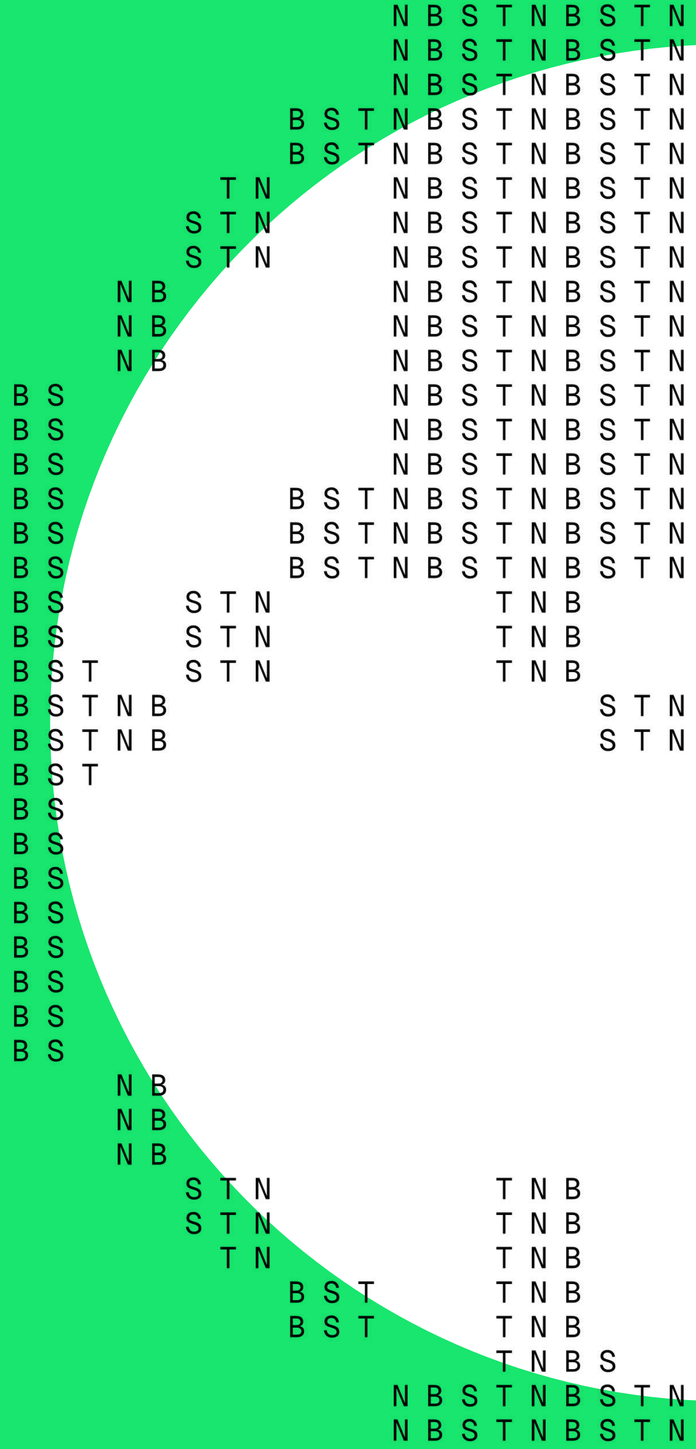
Why SGLang

- Highly performant inference runtime
- Day zero support for releases from Qwen, DeepSeek, and more
- Welcoming community and strong open-source ethos + enterprise adoption

Who uses SGLang?

Inference providers, foundation model labs, research institutions,
and AI product companies all use SGLang to power production workloads

History



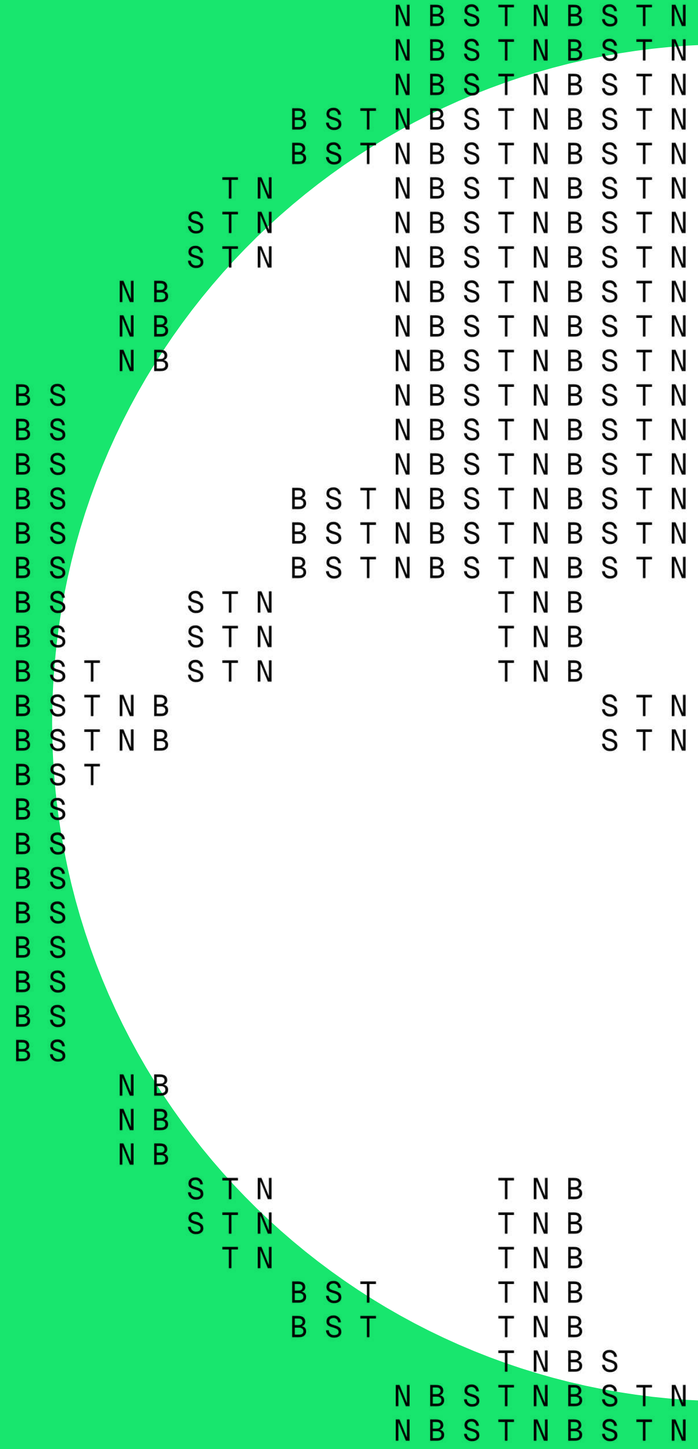
History of SGLang

- Dec 2023: Arxiv paper published
- Sep 2024: DeepSeek MLA release
- Oct 2024: First official meetup
- Dec 2024: DeepSeek-V3 support
- Jan 2025: Day-one DeepSeek-R1 support
- Mar 2025: Join PyTorch ecosystem
- June 2025: 15K GitHub stars (soon!)

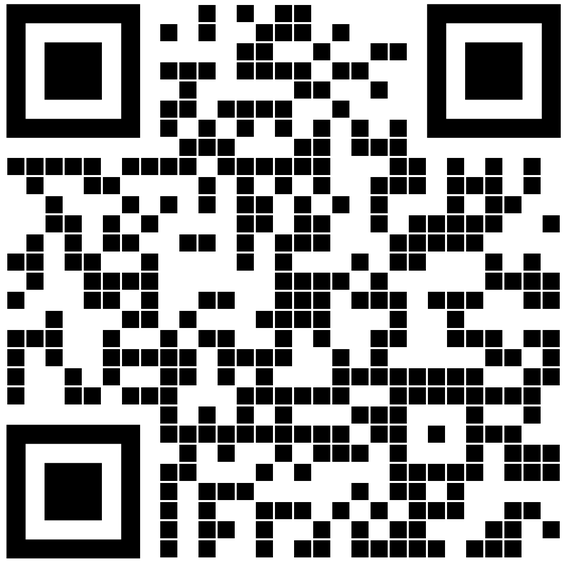
How I got involved with SGLang

- Work with Lianmin Zheng and Ying Sheng on SGLang
- Work with Zihao Ye on the FlashInfer project
- Core maintainer of SGLang with LMSYS Org

Setup



github.com/basetenlabs/SGLang-Workshop



Deploy your first model

- SGLang launch_server
- Bundle model weights in image
- Everything is a flag
- Our example: Llama 3.1 8B on L4

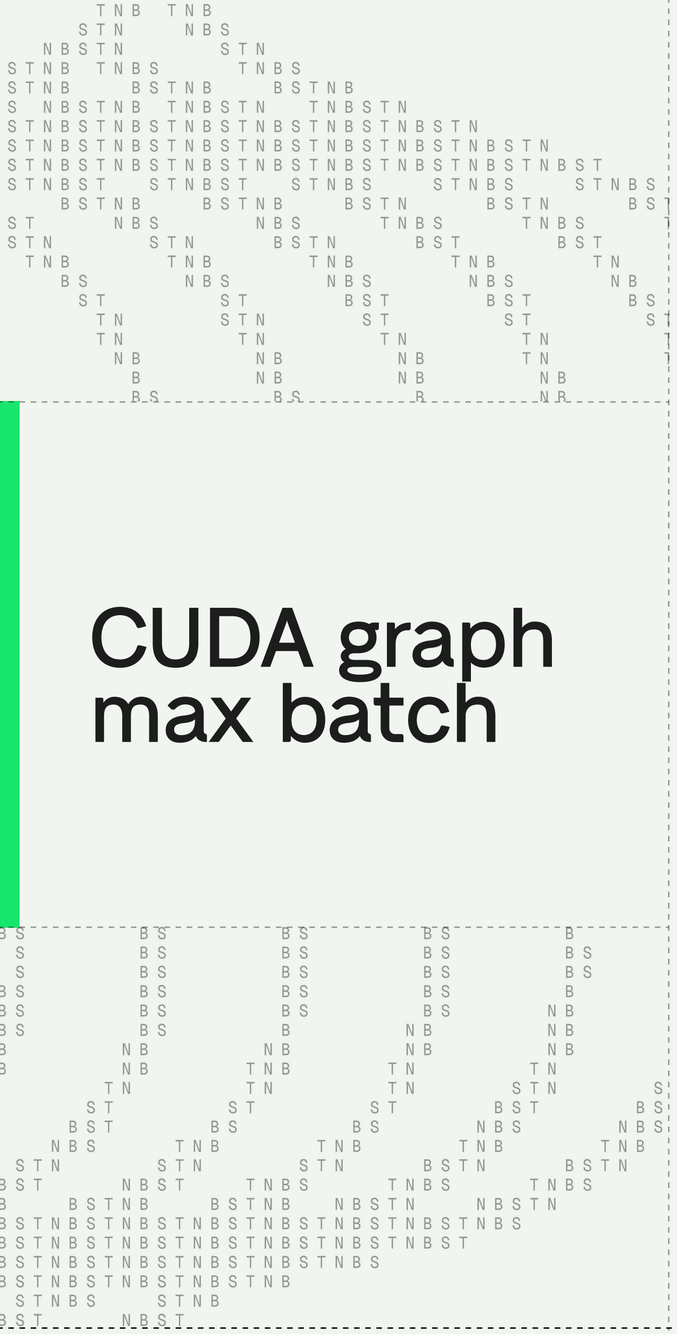
Quantization with FP8

- Use pre-built model weights
- Switch to FlashAttention backend for L4 compatibility
- Lovelace and Hopper FP8 support

Eagle 3 speculation

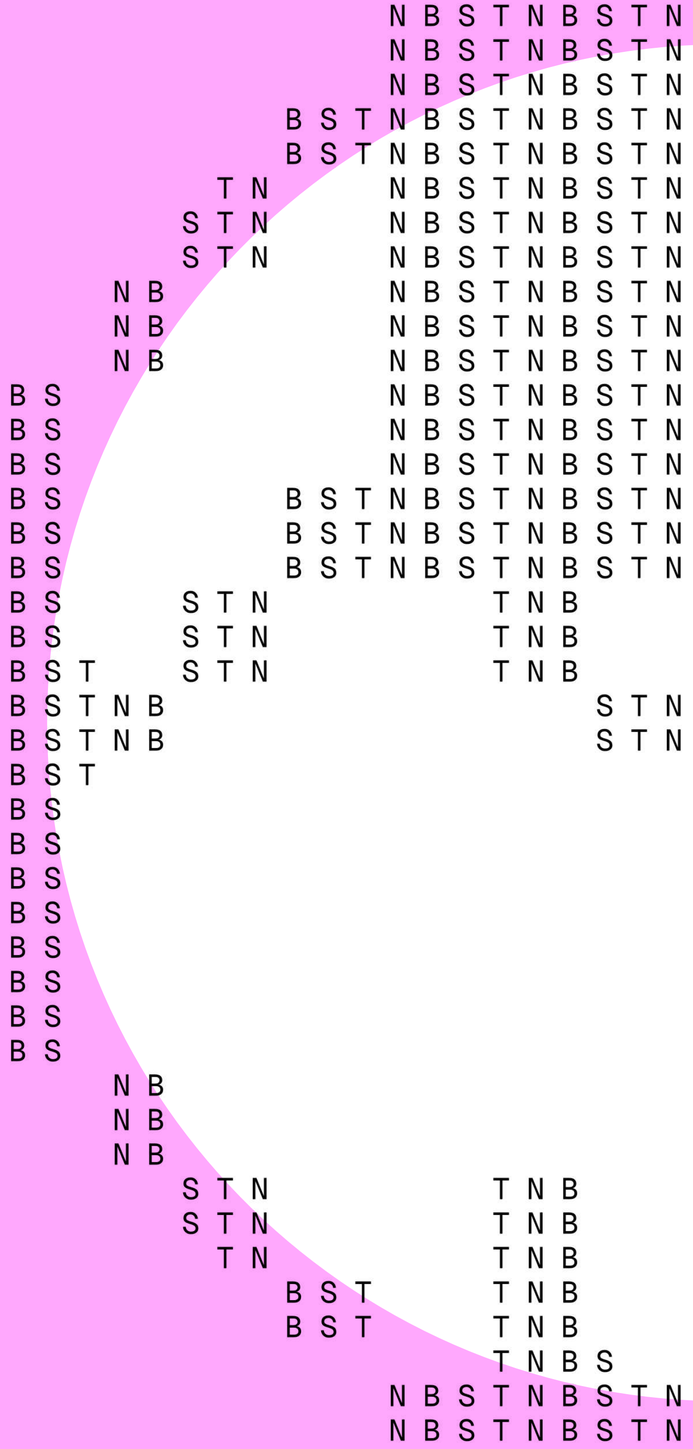
- SGLang supports many speculators
- EAGLE 3 creates draft model from target model layers
- Multi-layer fusion massively increases draft token acceptance rate

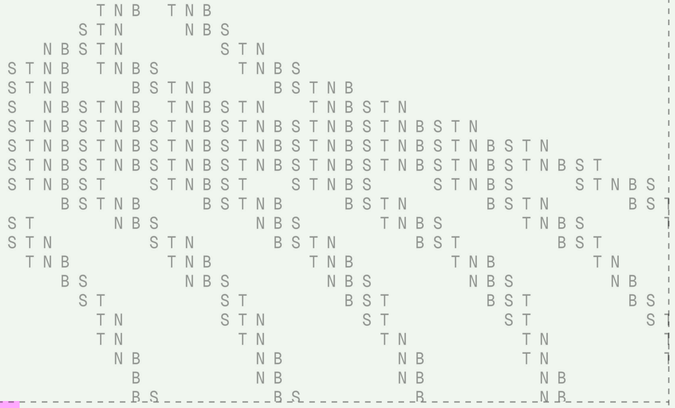
CUDA graph max batch



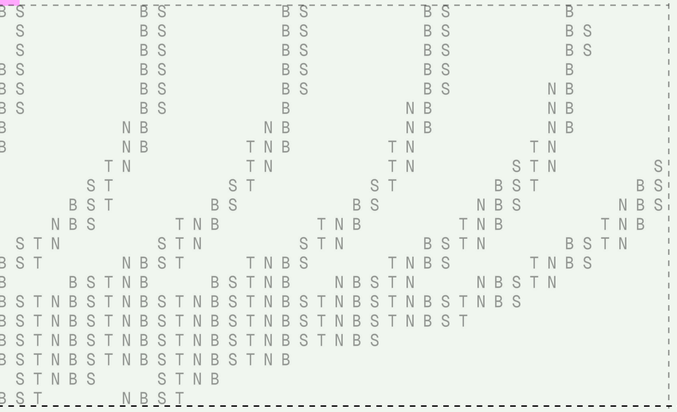
- By default, cuda-graph-max-bs is 8 on L4
- Updating to 32 improves perf massively
- We want CUDA graph to be True in decode

Community



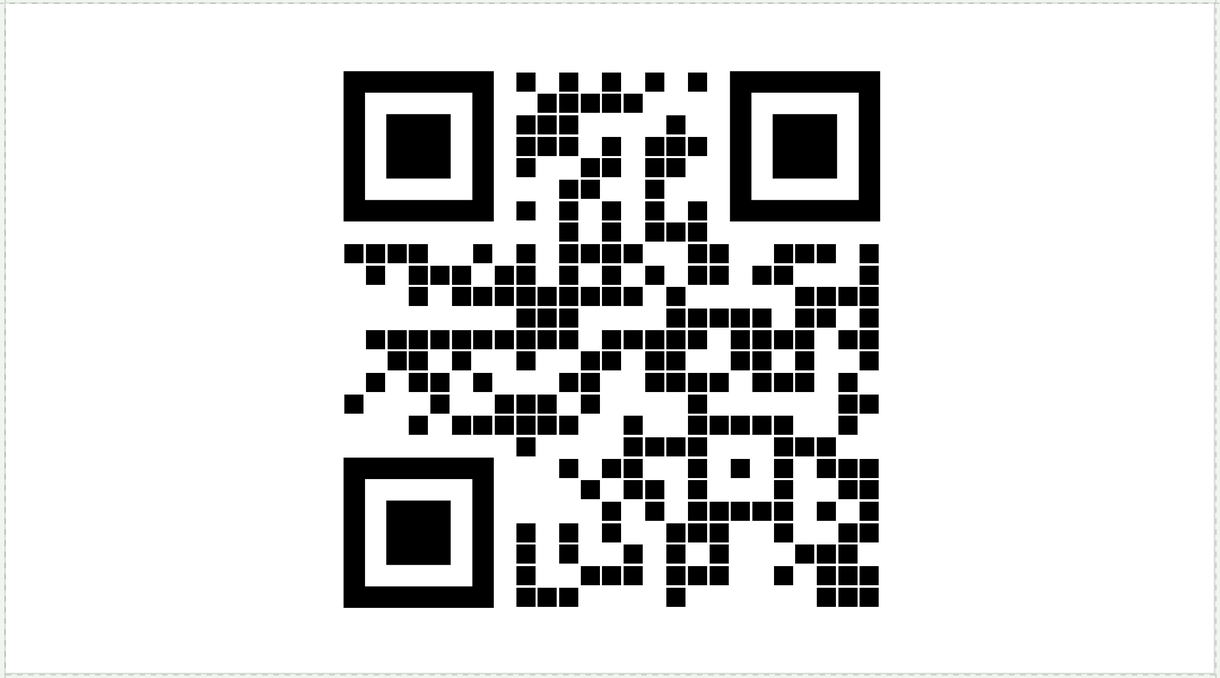


Join the community

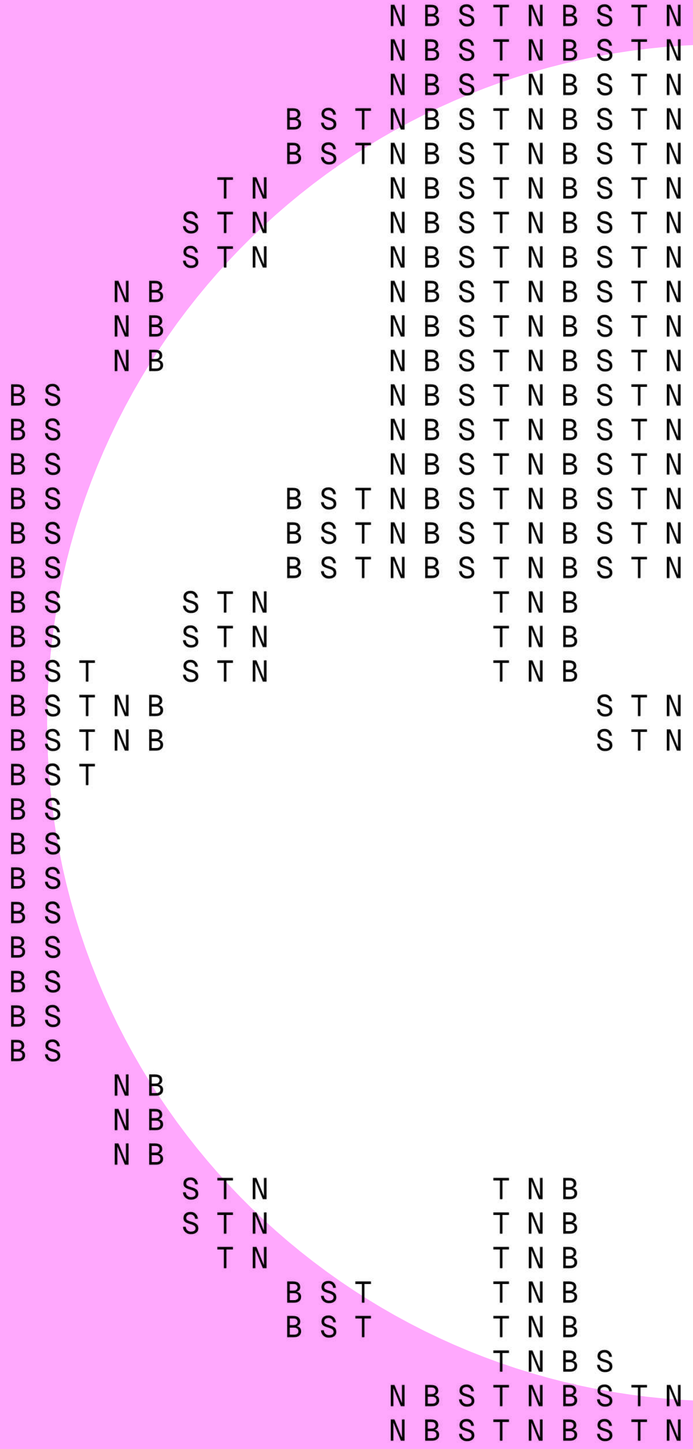


- Star SGLang on GitHub
- File issues and bug reports as you build
- Open your first PR! Good first issues are tagged.
- Join the Slack, follow on Twitter, and keep an eye out online + in-person meetups

SGLang community Slack

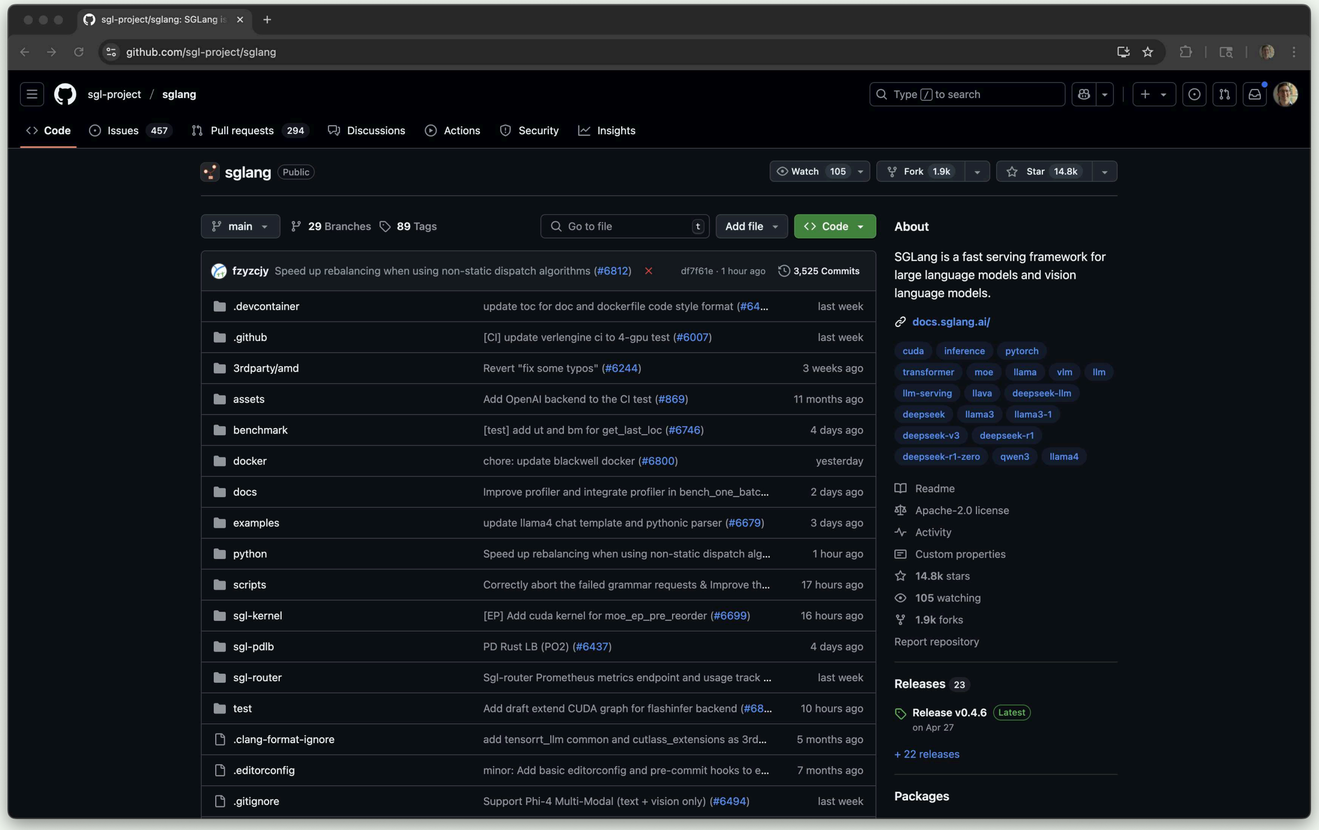


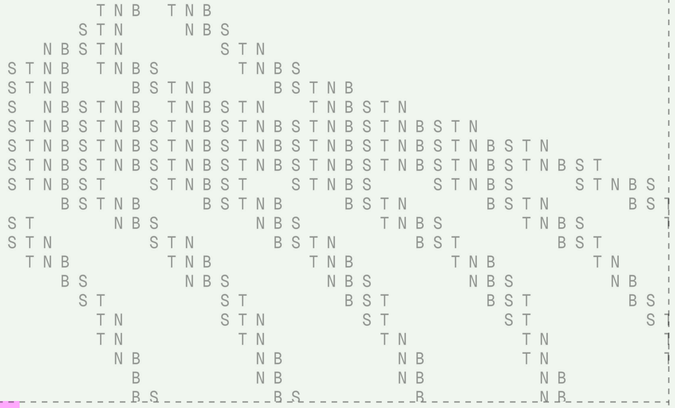
Codebase



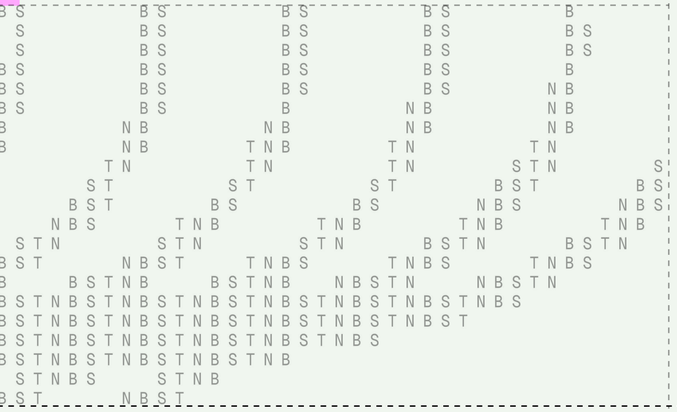
Tour of the codebase

SGLang has a clear architecture and is open to contributions.





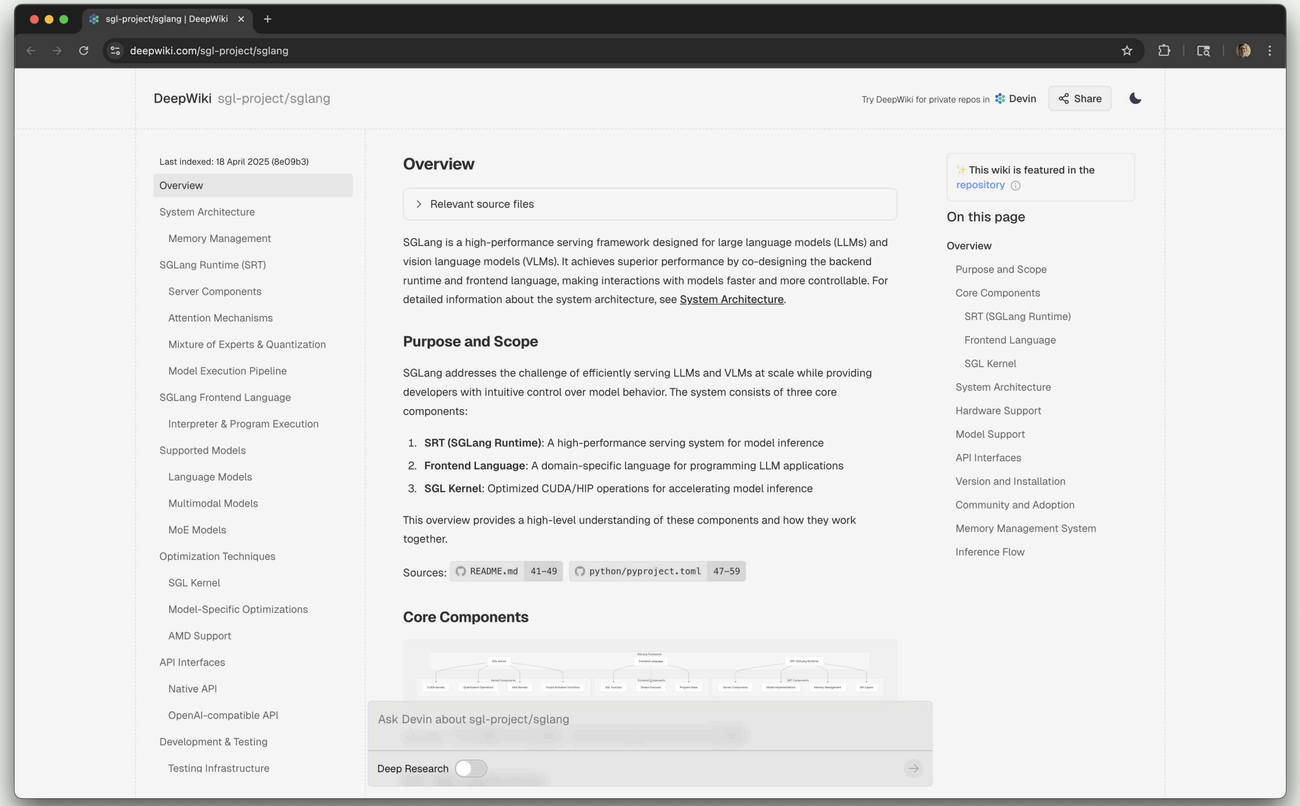
Tour of the codebase



1. **SRT (SGLang Runtime)**: A high-performance serving system for model inference
2. **Frontend Language**: A domain-specific language for programming LLM applications
3. **SGL Kernel**: Optimized CUDA/HIP operations for accelerating model inference

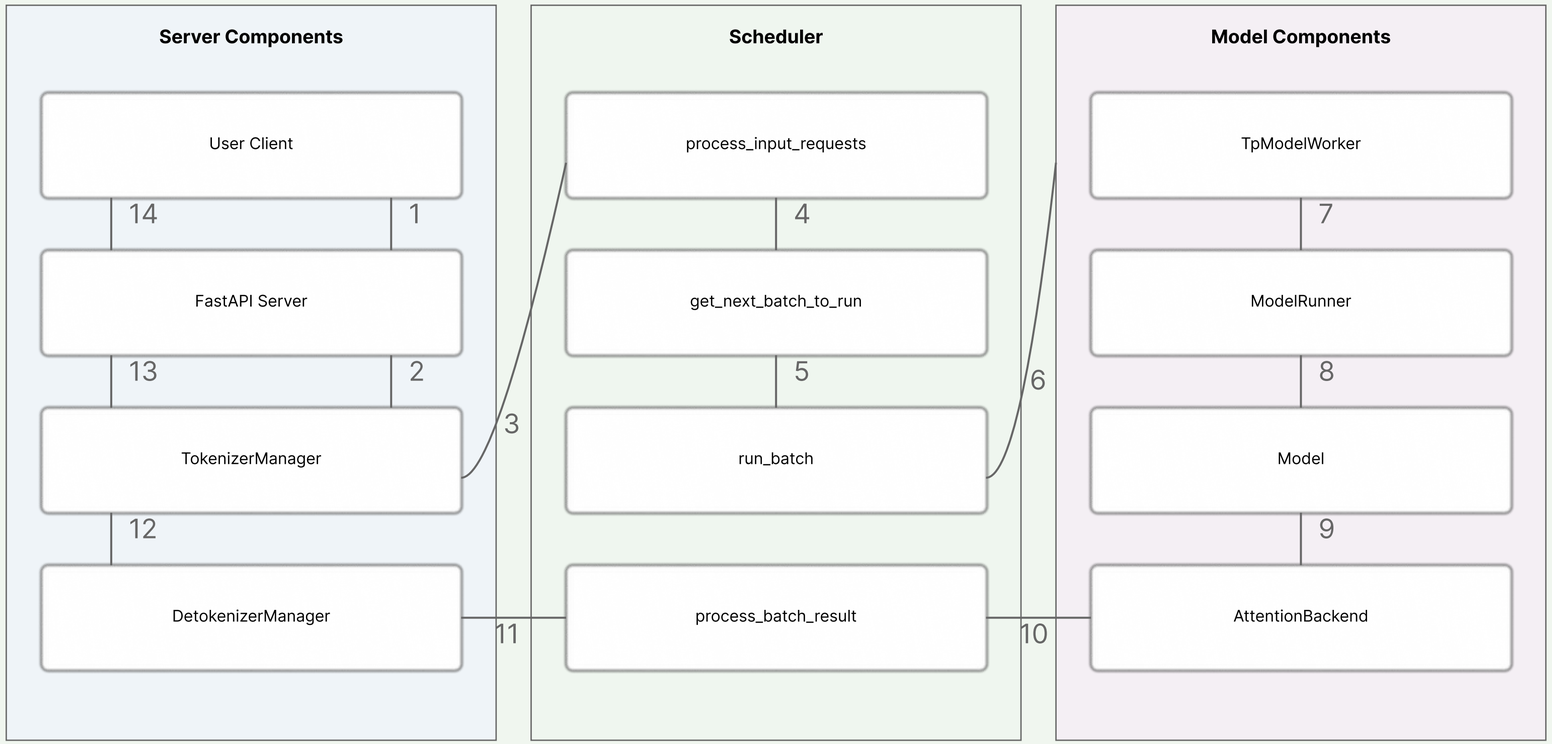
Tour of the codebase

https://deepwiki.com/sgl-project/sglang



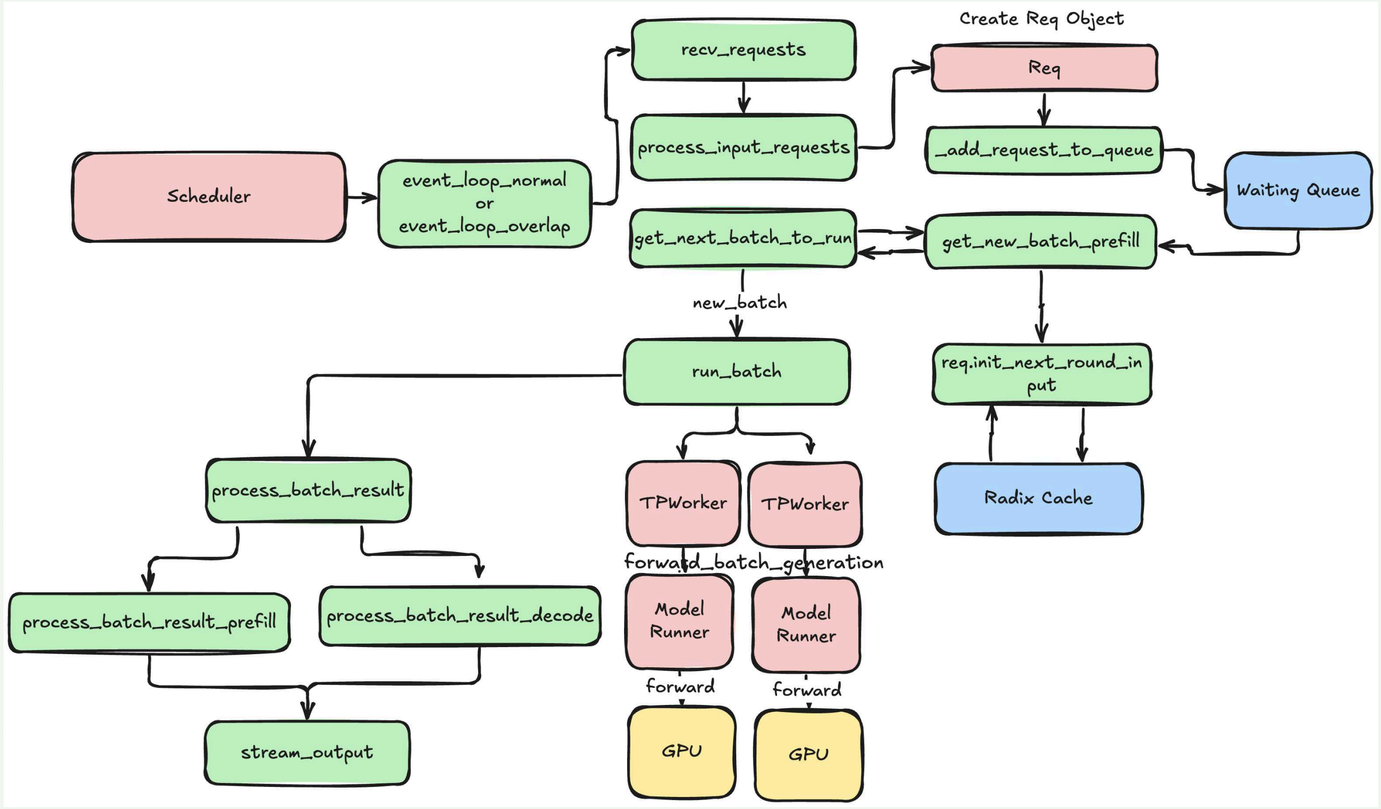
Request flow proces

Image from https://github.com/zhaochenyang20/Awesome-ML-SYS-Tutorial/

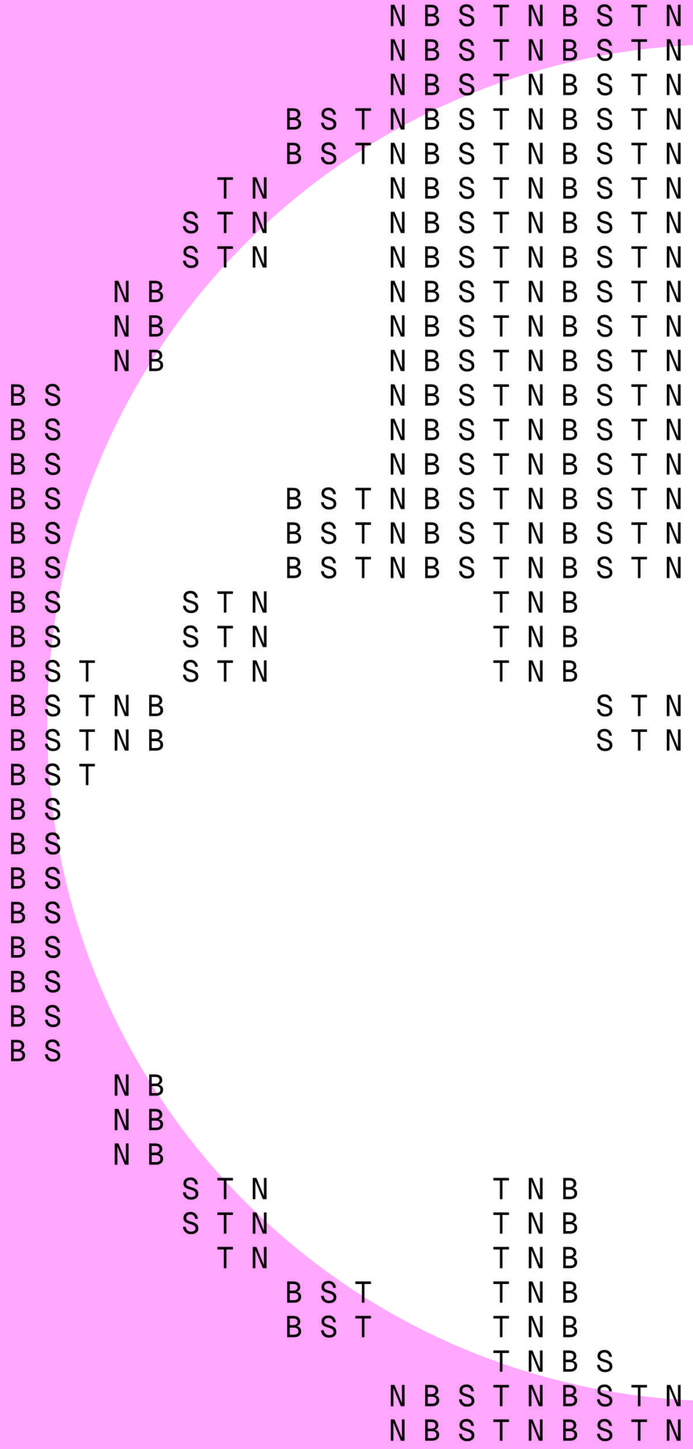


Scheduler

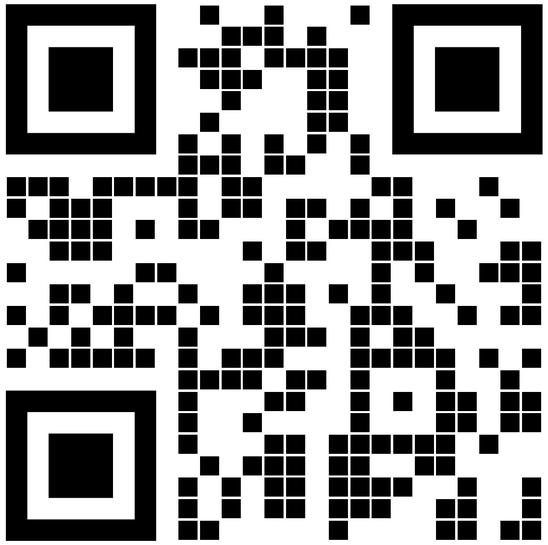
Image from https://github.com/zhaochenyang20/Awesome-ML-SYS-Tutorial/



Wrapping up



Join us Wednesday evening



Thank you

- x.com/basetenco
- linkedin.com/company/baseten

- x.com/zhyncs42
- linkedin.com/in/zhyncs

- x.com/philip_kiely
- linkedin.com/in/philipkiely