COMMUNITY & CITATION

Contributing to and Citing OpenSAI

Share reusable edge applications, deployment experience, documentation improvements, or core code—and cite OpenSAI accurately in academic research.

Overview | Quickstart | User Manual | Agent Integration | FAQ

Every improvement that helps the next developer or researcher succeed matters: clarifying a setup command or sharing a clean application package is just as valuable as adding a new scheduling algorithm.

1. Ways to Contribute

Your Background	High-Impact Contributions
Domain & AI Application Development	Reusable application packages (`app.yaml` + `steps/`), camera/LiDAR adapters, model wrappers, and calibrated knob profiles
Edge & Multi-Network Deployment	Field notes, startup scripts, and networking recipes for onboard compute units, roadside nodes, or multi-NIC servers
Core Runtime & Tooling	Enhancements to DAG placement, four-dimensional resource control, QUIC / `DELTA` multipath, SDK, CLI, or the Web console
Bilingual Docs & Agent Ecosystem	Clearer Chinese/English walkthroughs, MCP client configuration templates, and `SKILL.md` refinements
Reproducible Research & Evaluation	Well-documented workload evaluations with explicit hardware, network, and quality definitions

Submit focused bug fixes, typo corrections, and example improvements directly as Pull Requests. For larger changes affecting manifests, wire protocols, or scheduling behavior, open an Issue first outlining the user scenario, proposed design, and compatibility impact.

2. Reporting Reproducible Issues

Clear diagnostics help maintainers reproduce and resolve issues quickly. Please include:

- **Goal & minimal reproduction**: State what you intended to run and attach minimal commands, a small DAG JSON, or a compact application folder rather than an entire runtime directory.
- **Environment summary**: OpenSAI software version, operating system, Python version, online worker count, and relevant optional package versions.
- **Expected vs. actual behavior**: Include the failing step's error output, `job_id`, and `opensai explain` summary.
- **Performance context**: When discussing latency or adaptation choices, specify input size, CPU/GPU models, network conditions, repetition count, quality metric definition, and whether knobs were adaptive

or pinned with `-k`.

3. Packaging and Sharing an Application

A well-packaged OpenSAI application lets others run and adapt your workflow in minutes:

1. **Concise `README.md`**: Explain what the application does, its step graph, required Python packages, and how to read its output metrics.
2. **Validated execution**: Run `opensai app validate` and complete at least one end-to-end `opensai app run` in a clean environment before sharing.
3. **Zero-friction sample input**: When your full workflow relies on external datasets or large model weights, document where to obtain them and their license terms, and include a lightweight synthetic sample or generator script for immediate testing.
4. **Transparent knob definitions**: Document what each knob controls, how `quality` is computed, and the reference setup used to calibrate its tiers.

4. Code & Bilingual Documentation Standards

- **Keep changes focused**: Branch from the current public release line, address one well-defined behavior per PR, and summarize the motivation, user-facing effect, and verification steps.
- **Verify affected paths**: Exercise relevant runtime and CLI paths for backend changes; verify both **English** and **中文** UI views and live submissions for console changes; update both language versions of docs and examples whenever public flags, defaults, or manifest fields change.
- **Natural prose, exact technical parity**: Write idiomatic Chinese and English rather than word-for-word translations, while keeping CLI flags, environment variables, defaults, units (such as MiB vs. MB/s), and error codes strictly identical across languages.

5. Authorship & Open-Source Licensing

Only contribute material you have the right to share, and preserve original notices for any third-party components. OpenSAI is published under the MIT License; code and documentation contributed to the repository are released under the same MIT terms. Standalone external application packages you distribute independently may use any appropriate license declared in your package. Use your personal GitHub account when submitting commits so contributor attribution is preserved.

6. Citing OpenSAI in Academic Research

If OpenSAI materially supports your experiments, prototype system, or data-processing pipeline, please acknowledge OpenSAI in your paper's methodology or implementation section and cite the exact software version used (citing a versioned release rather than a moving branch improves reproducibility).

Recommended bibliography text:

> Institute on Networking Systems of AI (INSAI), Fudan University. *OpenSAI: Distributed DAG Computation Offloading and Bidirectional Adaptation Platform* (Version 1.0.0) [Computer software], 2026. https://github.com/INSAI-institute/OpenSAI

BibTeX entry:

```bibtex
@software{opensai2026,
  author       = {{Institute on Networking Systems of AI (INSAI), Fudan University}},
  title        = {{OpenSAI}: Distributed {DAG} Computation Offloading and Bidirectional Adaptation Platform},
  version      = {1.0.0},
  year         = {2026},
  url          = {https://github.com/INSAI-institute/OpenSAI},
  note         = {MIT License}
}
```

Machine-readable citation metadata is also included in `CITATION.cff` at the root of the code archive. When your work uses external models or datasets such as How2comm, V2XSet, or CIFAR-10, cite their original publications alongside OpenSAI.

7. Sharing Deployment Stories

We welcome real-world stories from cooperative perception, edge video analytics, industrial inspection, and classroom labs—including your device mix, workload scale, what saved the most engineering effort, and what should improve next (see the User Manual for full documentation; 中文贡献与学术引用说明).