

Contributing to Lycheetah × CASCADE

Thank you for your interest in contributing! This document provides guidelines for contributing to the project.

Current Status

This project is currently in **active development** (v2.0.1). We are focusing on:

1. Empirical validation with real data
2. Threshold optimization
3. Documentation improvements
4. Bug fixes and stability

How to Contribute

Reporting Issues

When reporting issues, please include:

- Python version and OS
- Complete error message/traceback
- Minimal code to reproduce the issue
- Expected vs actual behavior
- Your `lycheetah_config.py` settings if modified

Suggesting Enhancements

Enhancement suggestions are welcome! Please:

- Check existing issues first
- Explain the use case clearly
- Provide examples if possible
- Consider backwards compatibility

Code Contributions

Prerequisites

1. Fork the repository
2. Create a virtual environment:

```
bash

python -m venv venv
source venv/bin/activate # On Windows: venv\Scripts\activate
```

3. Install dependencies:

```
bash

pip install -e ".[dev]"
```

Development Workflow

1. **Create a branch:** `git checkout -b feature/your-feature-name`
2. **Make your changes**
3. **Follow code style:**

```
bash

black lycheetah/
flake8 lycheetah/
mypy lycheetah/
```

4. **Add tests:** All new features need tests
5. **Run test suite:**

```
bash

pytest tests/ -v --cov=lycheetah
```

6. **Update documentation:** If changing APIs or adding features
7. **Commit with clear messages:**

```
feat: add XYZ feature
fix: resolve issue with ABC
docs: update README with examples
test: add tests for feature XYZ
```

8. **Push and create PR**

Code Style Guidelines

- Use type hints for function signatures
- Write docstrings for all public functions/classes
- Keep functions under 50 lines when possible
- Use meaningful variable names
- Follow PEP 8 (enforced by black/flake8)
- Add comments for complex logic

Testing Guidelines

- Write unit tests for individual functions
- Write integration tests for workflows
- Aim for >80% code coverage
- Test edge cases and error conditions
- Use descriptive test names: `test_verify_provenance_with_empty_text`

Documentation Contributions

Documentation improvements are highly valued! You can:

- Fix typos or unclear explanations
- Add examples
- Improve API documentation
- Create tutorials or guides
- Translate documentation

✉ Research Contributions

If you're doing research with this system:

- Share validation results in discussions
- Propose improvements to metrics
- Submit threshold tuning data
- Cite the project in your papers

🚫 What We're NOT Looking For (Currently)

- Major architectural changes (discuss first in issues)

- Additional metrics without validation data
- Features that conflict with project goals
- Large PRs without prior discussion (>500 lines)

Commit Message Format

We use conventional commits:

```
<type>(<scope>): <subject>
```

```
<body>
```

```
<footer>
```

Types:

- `feat`: New feature
- `fix`: Bug fix
- `docs`: Documentation
- `style`: Formatting (no code change)
- `refactor`: Code restructuring
- `test`: Adding tests
- `chore`: Maintenance

Example:

```
feat(cascade_core): add phrase pattern detection
```

- Extend axiom detection beyond single words
- Add configurable phrase patterns
- Update tests to cover new functionality

```
Closes #42
```

Recognition

Contributors will be:

- Listed in CHANGELOG.md
- Acknowledged in release notes

- Added to a CONTRIBUTORS file
- Thanked in project documentation

Legal

By contributing, you agree that:

- Your contributions are your original work
- You have the right to submit the work
- Your contributions are licensed under MIT License
- You agree to the Signature Encoding Clause

Code of Conduct

Our Standards

- Be respectful and inclusive
- Accept constructive criticism gracefully
- Focus on what's best for the community
- Show empathy toward others
- No harassment, trolling, or discriminatory behavior

Enforcement

Violations should be reported to [your email]. All complaints will be reviewed and investigated. Project maintainers have the right to remove, edit, or reject contributions that don't align with this Code of Conduct.

Getting Help

- **Questions:** Open a GitHub Discussion
- **Bugs:** Open an Issue
- **Security:** See SECURITY.md
- **General:** Tag issues with `question` label

Project Roadmap

See [GitHub Projects](#) for current priorities and planned features.

Thank you for contributing to Lycheetah × CASCADE! 🌞

