# Contribution rules

- Kaolin Coding Style Guide can be found [here](STYLE\_GUIDE.md). We follow [Google C++ Style Guide](https://google.github.io/styleguide/cppguide.html) with few exceptions and additional guidelines regarding Kaolin-specific cases. See the Kaolin Coding Style Guide for details. When no rules can be found, follow the already occuring conventions. If there is no precedence in our codebase we are open to discussion.

- Avoid introducing unnecessary complexity into existing code so that maintainability and readability are preserved.

- Try to keep pull requests (PRs) as concise as possible:

- Avoid committing commented-out code.

- Wherever possible, each PR should address a single concern. If there are several otherwise-unrelated things that should be fixed to reach a desired endpoint, it is perfectly fine to open several PRs and state in the description which PR depends on another PR. The more complex the changes are in a single PR, the more time it will take to review those changes.

- Write PR and commit titles using imperative mood.

- Format commit messages sticking to rules described in [this](https://chris.beams.io/posts/git-commit/) guide.

- Make sure that the build log is clean, meaning no warnings or errors should be present.

- Some tests use licensed input datasets. If you don't have access to the necessary input datasets, you may skip those test(s), but please state in the PR comments which tests were not run.

- Kaolin's default build assumes recent versions of Kaolin's dependencies (CUDA, Pytorch, etc.). Contributions that add compatibility with older versions of those dependencies will be considered, but NVIDIA cannot guarantee that all possible build configurations work, are not broken by future contributions, and retain highest performance.

- Make sure that you can contribute your work to open source (no license and/or patent conflict is introduced by your code). You need to [`sign`](#Sign) your commit.

- Thanks in advance for your patience as we review your contributions; we do appreciate them!

<a name="Sign"></a>Sign your Work

--------------

We require that all contributors "sign-off" on their commits. This certifies that the contribution is your original work, or you have rights to submit it under the same license, or a compatible license.

Any contribution which contains commits that are not Signed-Off will not be accepted.

To sign off on a commit you simply use the `--signoff` (or `-s`) option when committing your changes:

$ git commit -s -m "Add cool feature."

This will append the following to your commit message:

Signed-off-by: Your Name <your@email.com>

By doing this you certify the below:

Developer Certificate of Origin

Version 1.1

Copyright (C) 2004, 2006 The Linux Foundation and its contributors.

1 Letterman Drive

Suite D4700

San Francisco, CA, 94129

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Developer's Certificate of Origin 1.1

By making a contribution to this project, I certify that:

(a) The contribution was created in whole or in part by me and I have the right to submit it under the open source license indicated in the file; or

(b) The contribution is based upon previous work that, to the best of my knowledge, is covered under an appropriate open source license and I have the right under that license to submit that work with modifications, whether created in whole or in part by me, under the same open source license (unless I am permitted to submit under a different license), as indicated in the file; or

(c) The contribution was provided directly to me by some other person who certified (a), (b) or (c) and I have not modified it.

(d) I understand and agree that this project and the contribution are public and that a record of the contribution (including all personal information I submit with it, including my sign-off) is maintained indefinitely and may be redistributed consistent with this project or the open source license(s) involved.