# Contributing

DeepSpeed welcomes your contributions!

## Prerequisites

DeepSpeed uses [pre-commit](https://pre-commit.com/) to ensure that formatting is

consistent across DeepSpeed. First, ensure that `pre-commit` is installed from either

installing DeepSpeed or `pip install pre-commit`. Next, the pre-commit hooks must be

installed once before commits can be made:

```bash

pre-commit install

```

Afterwards, our suite of formatting tests run automatically before each `git commit`. You

can also run these manually:

```bash

pre-commit run --all-files

```

If a formatting test fails, it will fix the modified code in place and abort

the `git commit`. After looking over the changes, you can `git add <modified files>`

and then repeat the previous `git commit` command.

## Testing

DeepSpeed tracks two types of tests: unit tests and more costly model convergence tests.

The model convergence tests train

[DeepSpeedExamples](https://github.com/microsoft/DeepSpeedExamples/) and measure

end-to-end convergence and related metrics. Unit tests are found in `tests/unit/` and

the model convergence tests are found in `tests/model/`.

### Unit Tests

[PyTest](https://docs.pytest.org/en/latest/) is used to execute tests. PyTest can be

installed from PyPI via `pip install pytest`. Simply invoke `pytest --forked` to run the

unit tests:

```bash

pytest --forked tests/unit/

```

You can also provide the `-v` flag to `pytest` to see additional information about the

tests. Note that [pytest-forked](https://github.com/pytest-dev/pytest-forked) and the

`--forked` flag are required to test CUDA functionality in distributed tests.

### Model Tests

To execute model tests, first [install DeepSpeed](#installation). The

[DeepSpeedExamples](https://github.com/microsoft/DeepSpeedExamples/) repository is cloned

as part of this process. Next, execute the model test driver:

```bash

cd tests/model/

pytest run\_sanity\_check.py

```

Note that the `--forked` flag is not necessary for the model tests.

## Contributor License Agreement

This project welcomes contributions and suggestions. Most contributions require you to

agree to a Contributor License Agreement (CLA) declaring that you have the right to, and

actually do, grant us the rights to use your contribution. For details, visit

https://cla.opensource.microsoft.com.

When you submit a pull request, a CLA bot will automatically determine whether you need

to provide a CLA and decorate the PR appropriately (e.g., status check, comment). Simply

follow the instructions provided by the bot. You will only need to do this once across

all repos using our CLA.

## Code of Conduct

This project has adopted the [Microsoft Open Source Code of

Conduct](https://opensource.microsoft.com/codeofconduct/). For more information see the

[Code of Conduct FAQ](https://opensource.microsoft.com/codeofconduct/faq/) or contact

[opencode@microsoft.com](mailto:opencode@microsoft.com) with any additional questions or

comments.