

• Code Style Check

Author: Hongzhi Chen

Last Update: Oct 8, 2022

- **Why do we need to write high quality code with consistent style?**
 - The code is written once but will be read multiple times in the future.
- **Conventions in DGL:** [DGL Code Conventions](#)

TL;DR

- Reformat Python
 - `gtests/python/`
- Reformat C++
 - `clang-format -i filename`
- Take advantage of github linrunner
 - `linrunner -a`

Linrunner: a linter management tool

We're using linrunner to manage linters and formatters in DGL.

See <https://github.com/pytorch/pytorch/wiki/linrunner> for the usage.

Python

ufmt (Black + usort)

ufmt is a unified formatter consisting of Black and usort (an alternative to isort and is fully compatible with Black).

- Install ufmt
 - `pip install -U ufmt`
- Commands:
 - `check` Check formatting of one or more paths
 - `diff` Generate diffs for any files that need formatting
 - `format` Format one or more paths in place
- Example

- `ufmt format tests/python/`

Black (Not recommended, use ufmt instead)

Black can help you automatically reformat your python code to be more compliant with our code style, but **it does not solve all the problems**, gradually familiarizing yourself with our [python style guide](#).

- Install black
 - `pip install --upgrade black`
- Then run the command from the dgl/ dir. ⚠️⚠️ **Warning:** Before you run the command, it is highly suggested to commit your change, such that if the black crashes your work, you can simply run `git reset --hard` to revert all the change.
 - `black filename`
- If auto format causes unexpected problems that can't be reasonably resolved, exclude it in `pyproject.toml`.

Flake8

Flake8 is a tool to help you check potential pitfalls in your python code, and provide warnings.

TODO(Steve): [integrate](#) flake8 with Github PR reviewer.

- Install Flake8 (it should be already installed, if not):
 - `pip install --upgrade flake8`
- Then run the command from the dgl/ dir
 - `flake8 --max-line-length=80 filename`
- You will get warnings like this:

```
python/dgl/utils/pin_memory.py:52:1: E302 expected 2 blank
lines, found 1
python/dgl/utils/pin_memory.py:53:80: E501 line too long (86 >
79 characters)
python/dgl/utils/pin_memory.py:65:5: F821 undefined name
'_CAPI_DGLIndexScatterGPUCPU'
```

```
python/dgl/utils/pin_memory.py:66:9: E128 continuation line
under-indented for visual indent
```

Isort (Not recommended, use ufmt instead)

Isort is a tool to help you sort your import.

TODO(Steve): create one script to run all auto formatters.

- Install isort
 - `pip install --upgrade isort`
- Then run the command from the dgl/ dir.  **Warning:** Before you run the command, it is highly suggested to commit your change, such that if the isort crashes your work, you can simply run `git reset` to revert all the change.
 - `isort filename`

Autopep8 (Not recommended, use ufmt instead)

Autopep8 can help you automatically reformat your python code to be more compliant with our code style, but **it does not solve all the problem**, gradually familiarizing yourself with our [python style guide](#).

- Install Autopep8
 - `pip install --upgrade autopep8`
- Create pyproject.toml in the dir you run the command with the follow content
 - **TODO(Steve):** check-in the file to github later

```
[tool.autopep8]
max_line_length = 80
in-place = true
aggressive = 3
# Add the path to here if you want to exclude them from
autopep8 auto reformat.
exclude = '''
    .github/*,
    build/*,
    '''
```

- Then run the command from the dgl/ dir.   **Warning:** Before you run the command, it is highly suggested to commit your change, such that if the autopep8 crashes your work, you can simply run `git reset` to revert all the change.
 - `autopep8 filename`
- Or run the following command directly without the file
 - `autopep8 --in-place --aggressive --aggressive --max-line-length=80 filename`
- If auto format causes unexpected problems that can't be reasonably resolved, exclude it in pyproject.toml.

C++

ClangFormat

ClangFormat can help you automatically reformat your c++ code to be more compliant with our code style, but **it does not solve all the problems**, gradually familiarizing yourself with our [C++ style guide](#) (Google).

- Install ClangFormat
 - `pip install --upgrade clang-format`
- Then run the command from the dgl/ dir.   **Warning:** Before you run the command, it is highly suggested to commit your change, such that if the clang-format crashes your work, you can simply run `git reset --hard` to revert all the change.
 - `clang-format -i filename`

Appendix

-  DGL Code Conventions
- Learn more about black
 - <https://black.readthedocs.io/en/stable/>
- Learn more about flake8
 - <https://flake8.pycqa.org/en/latest/>
- Learn more about isort
 - <https://pycqa.github.io/isort/>
- Learn more about Autopep8

- <https://github.com/hhatto/autopep8>
- Learn more about ClangFormat
 - <https://clang.llvm.org/docs/ClangFormat.html>