# Introduction to Research

## (3) Academic Collaborations

**Rex Ying**

# Research Collaborations

- Help each other in their research
  - Ideas, research discussions
  - Code, figures, references
  - Recognize that our individual strengths could be complementary
- I encourage research collaborations among lab members, as well as other research partners
- **Co-authorship**: significant contribution to the paper
  - Major effort in writing (e.g., a few sections)
  - Experiments: ablation, baselines, additional features in contribution
  - Discuss co-authorship at least 2 months before deadline

# Overview

- Within computer science
- Inter-disciplinary collaborations
- Industry collaborations

# Within Computer science

- Opportunities for graph learning to benefit from a variety of CS techniques and vice versa
  - Graph+NLP/vision research
  - Graph for RL; RL for graphs
  - System acceleration for specific graph learning problems
  - HCI? Graphics? Theory? …
  - It's up to you to find opportunities to collaborate (talk to other students and professors!)

# GitHub Repository

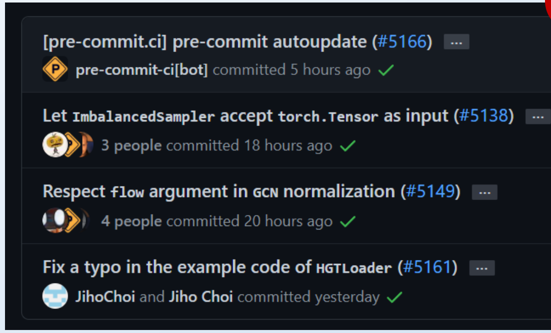
- https://github.com/Graph-and-Geometric-Learning
  - Create a repository, and have a comprehensive Readme
  - Experiment setup; commands; Arguments; Datasets; FAQ …

- Answer GitHub issues; questions from emails
- Make updates to the code if necessary (especially after rebuttal)

**Maintaining and enforcing good coding practices can greatly accelerate collaboration!**

# Engineering Practice

- Good coding styles (more in future session)
- Use versioning
  - Revert when there is a bug
- Use pull requests
- Github actions; unit tests
- Record experiment settings
  - Use AutoML / experiment management tools
- Keep **visualization code** and Jupyter notebooks
  - [MLflow](), Tensorboard
- Use documentation (docstring)!

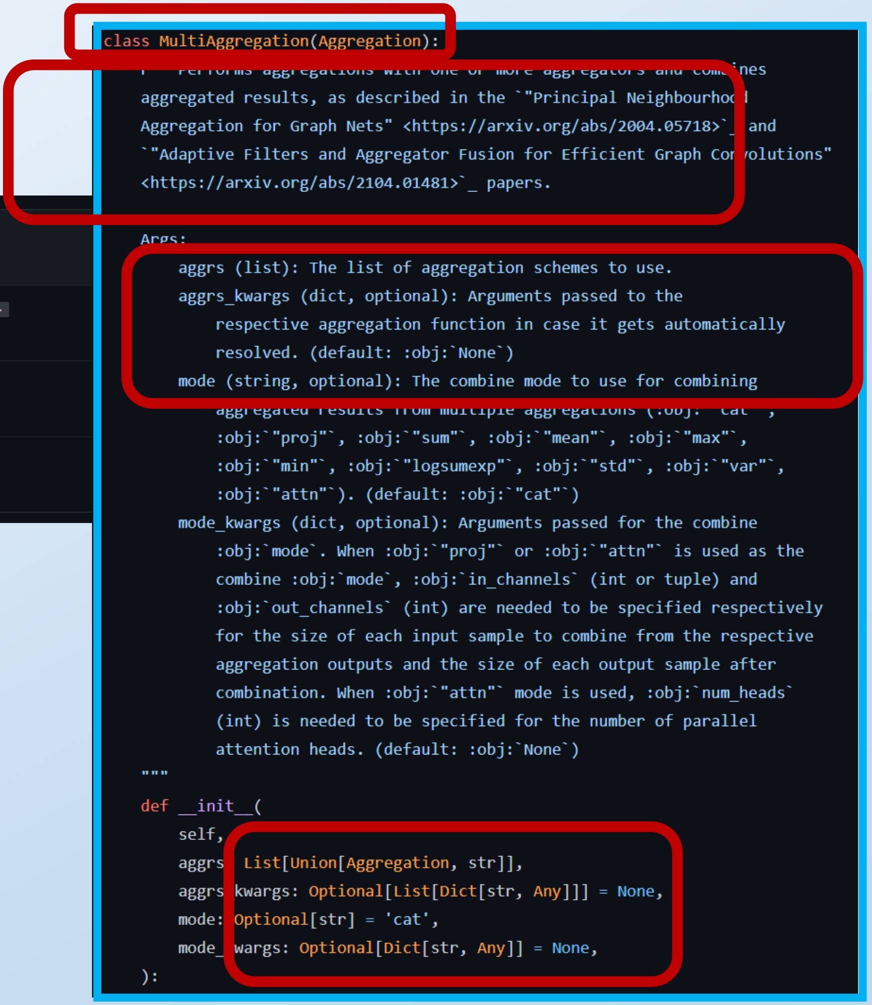


[pre-commit.ci] pre-commit autoupdate (#5166) ...
pre-commit-ci[bot] committed 5 hours ago ✓

Let `ImbalancedSampler` accept `torch.Tensor` as input (#5138) ...
3 people committed 18 hours ago ✓

Respect `flow` argument in `GCN` normalization (#5149) ...
4 people committed 20 hours ago ✓

Fix a typo in the example code of `HGTLoader` (#5161) ...
JihoChoi and Jiho Choi committed yesterday ✓



```
class MultiAggregation(Aggregation):
    r"""Performs aggregations with one or more aggregators and combines
    aggregated results, as described in the `"Principal Neighbourhood
    Aggregation for Graph Nets" <https://arxiv.org/abs/2004.05718>`_ and
    `"Adaptive Filters and Aggregator Fusion for Efficient Graph Convolutions"
    <https://arxiv.org/abs/2104.01481>`_ papers.

    Args:
        aggrs (list): The list of aggregation schemes to use.
        aggrs_kwargs (dict, optional): Arguments passed to the
            respective aggregation function in case it gets automatically
            resolved. (default: :obj:`None`)
        mode (string, optional): The combine mode to use for combining
            aggregated results from multiple aggregations (:obj:`"cat"`,
            :obj:`"proj"`, :obj:`"sum"`, :obj:`"mean"`, :obj:`"max"`,
            :obj:`"min"`, :obj:`"logsumexp"`, :obj:`"std"`, :obj:`"var"`,
            :obj:`"attn"`). (default: :obj:`"cat"`)
        mode_kwargs (dict, optional): Arguments passed for the combine
            :obj:`mode`. When :obj:`"proj"` or :obj:`"attn"` is used as the
            combine :obj:`mode`, :obj:`in_channels` (int or tuple) and
            :obj:`out_channels` (int) are needed to be specified respectively
            for the size of each input sample to combine from the respective
            aggregation outputs and the size of each output sample after
            combination. When :obj:`"attn"` mode is used, :obj:`num_heads`
            (int) is needed to be specified for the number of parallel
            attention heads. (default: :obj:`None`)
    """

    def __init__(
        self,
        aggrs: List[Union[Aggregation, str]],
        aggrs_kwargs: Optional[List[Dict[str, Any]]] = None,
        mode: Optional[str] = 'cat',
        mode_kwargs: Optional[Dict[str, Any]] = None,
    ):
```

# Interdisciplinary Collaboration

- Talk to PhDs and professors from different departments
- When you take class, always have your research in mind

- Graph and geometric learning is an area **that benefits tremendously from real-world applications**!
  - Many people don’t realize that their problems can be formulated as graph learning

# Collaboration with Industry

- Typically, you want 1 **recommendation** from industry
- Internship is a great way to show people your research ability and get recommendation!
- Yale allows **2 internships** as PhD students
  - 1 internship in 1st or 2nd year
  - 1 internship in 3rd or 4th year
    - Deepmind and some other companies usually require at least 3rd year
- Involve me during your internship, to make sure that you work on meaningful projects and produce good research

# Authorship (1)

- We encourage collaborations
- But we should have a uniform **standard on authorship**
  - **First author** leads the project (majority of implementations; paper writing; leads research discussions)
  - **Co-authors** should have a **significant contribution** to the paper
    - Substantial coding (*e.g.*, method implementation; baseline implementation)
    - Write at least 1 major section; and actively participate in writing, polishing
    - In charge of dataset processing and preparation
    - Participate in discussions and propose ideas
    - Major contribution in refactoring; website; posters; slides
    - **At least 2 (if not more) of the above!**
  - Acknowledgement: one of the contributions above

# Authorship (2)

- It’s ok and necessary to politely discuss authorship
  - Postponing it too much can lead to unpleasant conversations
  - Set people’s expectation early
  - **No** addition of authorship close to the submission deadline

- Please talk about authorship **at least 2 months** before submission!!
  - So people are aligned on what level of efforts should be made for the project
  - Can be discussed earlier and multiple times as long-term projects progress

# Summary

- Impact is a multi-dimensional goal, and so is collaboration
  - **Within computer science**: techniques and problems from other similar fields
  - **Inter-disciplinary collaboration**: create a larger impact
  - **Tool-building**: high-quality code, libraries, benchmarks (for the research community and beyond)
- Co-authorship discussion
- Code quality and engineering practices