

# HLPW6: How to upload results to GitHub

Daniel Heathcote

[Heathcote.Daniel@aurora.aero](mailto:Heathcote.Daniel@aurora.aero)

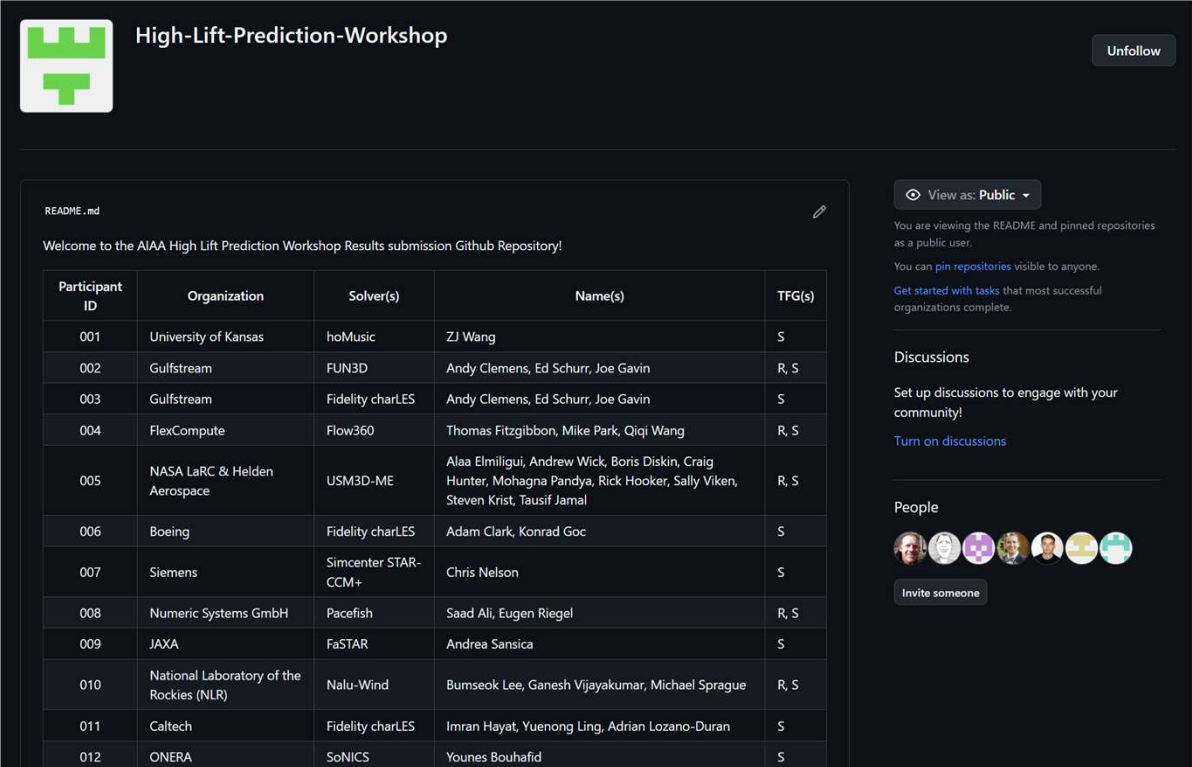
Rev A, 1<sup>st</sup> April 2026

# Overview

- For HLPW6, a new method of collecting results is being used: via GitHub
- This method has been successfully used by DPW
- This document outlines a method by which submission files can be uploaded
- It is assumed that the submitter already has a GitHub account
  - If not, sign up is free via [www.github.com](http://www.github.com)
- It is also assumed that all operations are completed in-browser, alternate workflows may be available via command line or desktop application

# Accessing the organization

- HLPW6 GitHub:
  - [High-Lift-Prediction-Workshop](#)
- Welcome page should display participant information & IDs
- If your name isn't present and you wish to submit results, please reach out to TFG leads for an ID



High-Lift-Prediction-Workshop

Unfollow

README.md

Welcome to the AIAA High Lift Prediction Workshop Results submission Github Repository!

Participant ID	Organization	Solver(s)	Name(s)	TFG(s)
001	University of Kansas	hoMusic	ZJ Wang	S
002	Gulfstream	FUN3D	Andy Clemens, Ed Schurr, Joe Gavin	R, S
003	Gulfstream	Fidelity charLES	Andy Clemens, Ed Schurr, Joe Gavin	S
004	FlexCompute	Flow360	Thomas Fitzgibbon, Mike Park, Qiqi Wang	R, S
005	NASA LaRC & Helden Aerospace	USM3D-ME	Alaa Elmiligui, Andrew Wick, Boris Diskin, Craig Hunter, Mohagna Pandya, Rick Hooker, Sally Viken, Steven Krist, Tausif Jamal	R, S
006	Boeing	Fidelity charLES	Adam Clark, Konrad Goc	S
007	Siemens	Simcenter STAR-CCM+	Chris Nelson	S
008	Numeric Systems GmbH	Pacefish	Saad Ali, Eugen Riegel	R, S
009	JAXA	FaSTAR	Andrea Sansica	S
010	National Laboratory of the Rockies (NLR)	Nalu-Wind	Bumseok Lee, Ganesh Vijayakumar, Michael Sprague	R, S
011	Caltech	Fidelity charLES	Imran Hayat, Yuenong Ling, Adrian Lozano-Duran	S
012	ONERA	SoNICS	Younes Bouhafid	S

View as: Public

You are viewing the README and pinned repositories as a public user.

You can pin repositories visible to anyone.

Get started with tasks that most successful organizations complete.

Discussions

Set up discussions to engage with your community!

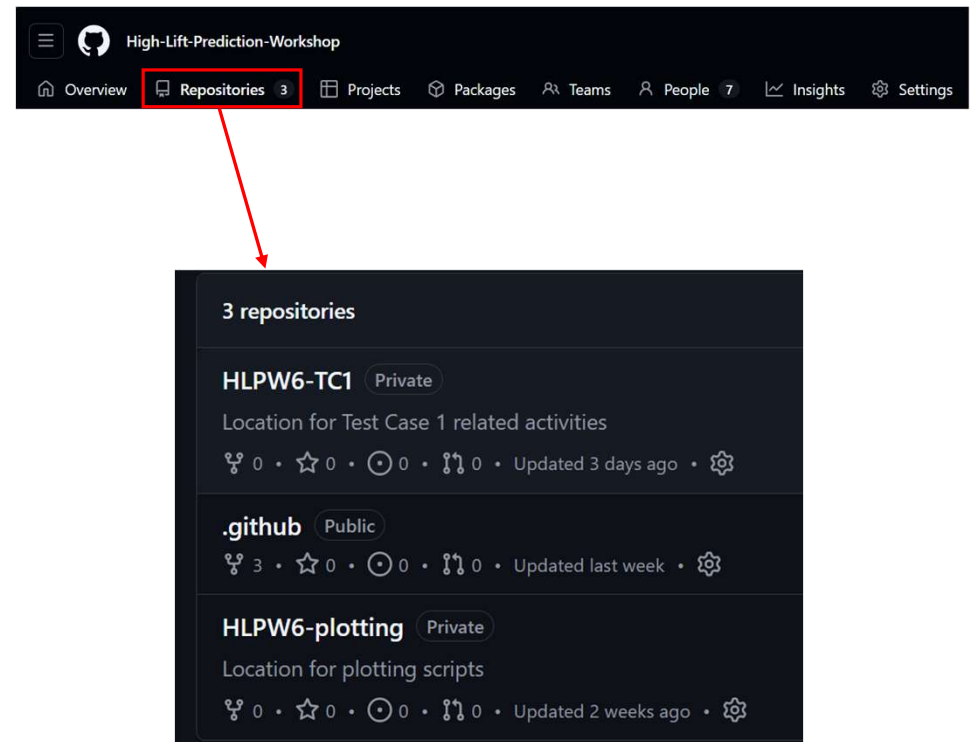
Turn on discussions

People

Invite someone

# Repository Structure

- The organization currently contains three repositories:
  - **HLPW6-TC1**: Repo for test case 1 submissions
  - **.github**: do not open/modify, relates to participant table
  - **HLPW6-plotting**: Associated tecplot macros, layout files and scripts for plotting and comparing results
- Navigate to HLPW6-TC1:
  - [High-Lift-Prediction-Workshop/HLPW6-TC1: Location for Test Case 1 related activities](#)



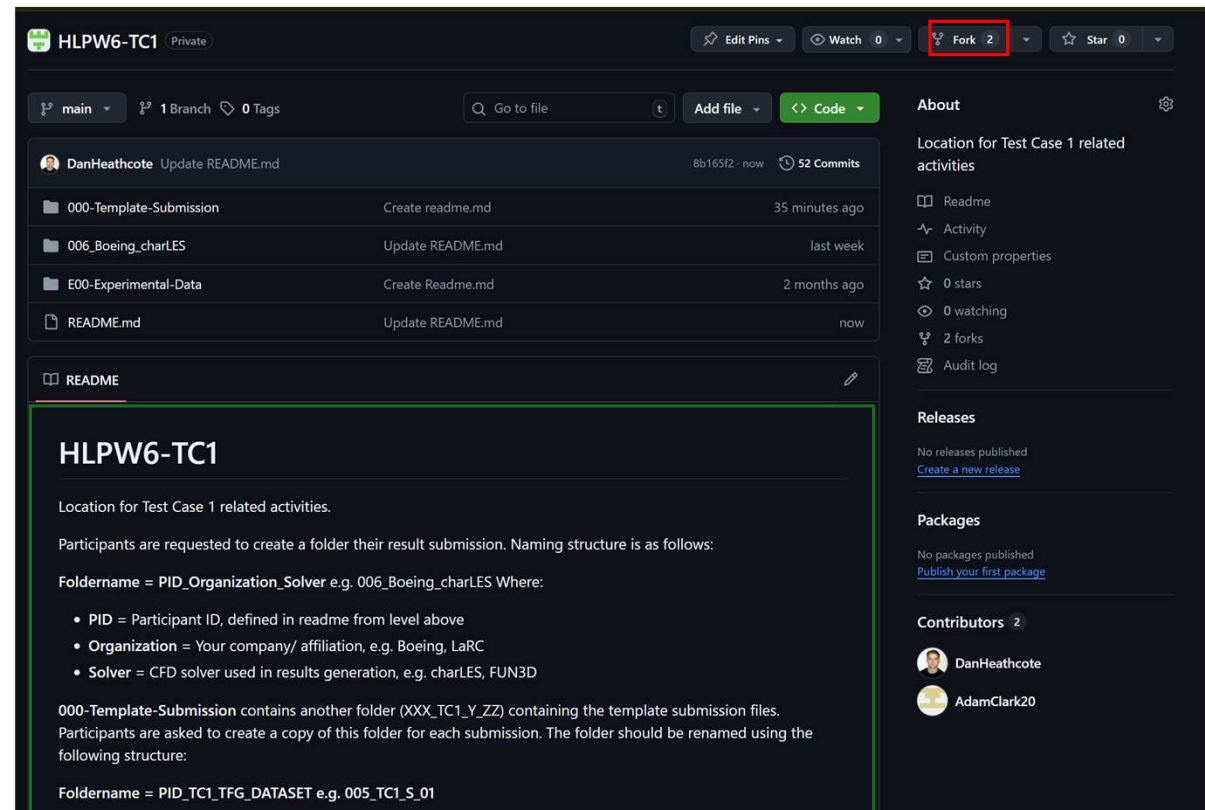
# Repo/ Submission Folder Structure

- **HLPW6 – TC1 Repo**

- **000-Template-Submission:** Replace 000 with your participant ID
  - **000\_TC1\_Y\_SS:** Corresponds to a specific submission to a specific TFG. Template files are located here
    - **FlowViz:** location for flow visualization images. Utilize standard Tecplot layout file containing requisite views as defined by TFG leads and RANS visualization subcommittee
- **E00-Experimental Data: placeholder for experimental data**

# Creating your HLPW6-TC1 fork (1/2)

- Creating a **fork** of the repo keeps any changes you make to the repo isolated until they are **“pushed”** to the main branch.
- This is done via **forking (red highlight)**
- You will use this branch to upload and submit your data
- Please read the contents of the **README.md** file carefully, as these provide the guidelines for naming submission folders



The screenshot shows the GitHub interface for the repository 'HLPW6-TC1'. The 'Fork' button is highlighted with a red box. The repository has 2 forks and 0 stars. The README file is open, showing the following content:

```
HLPW6-TC1

Location for Test Case 1 related activities.

Participants are requested to create a folder their result submission. Naming structure is as follows:

Foldername = PID_Organization_Solver e.g. 006_Boeing_charLES Where:

• PID = Participant ID, defined in readme from level above
• Organization = Your company/ affiliation, e.g. Boeing, LaRC
• Solver = CFD solver used in results generation, e.g. charLES, FUN3D

000-Template-Submission contains another folder (XXX_TC1_Y_ZZ) containing the template submission files.
Participants are asked to create a copy of this folder for each submission. The folder should be renamed using the following structure:

Foldername = PID_TC1_TFG_DATASET e.g. 005_TC1_S_01
```

# Creating your HLPW6-TC1 fork (2/2)

- Creating a fork places a copy of the “**main**” branch in your GitHub
- Please create your own **fork** by clicking the highlighted button.
- The fork’s name can be anything you fancy, although keeping it the same as the master is recommended

## Create a new fork

A *fork* is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project.

Required fields are marked with an asterisk (\*).

Owner \*  / Repository name \*

HLPW6-TC1 is available.

By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

Description

43 / 350 characters

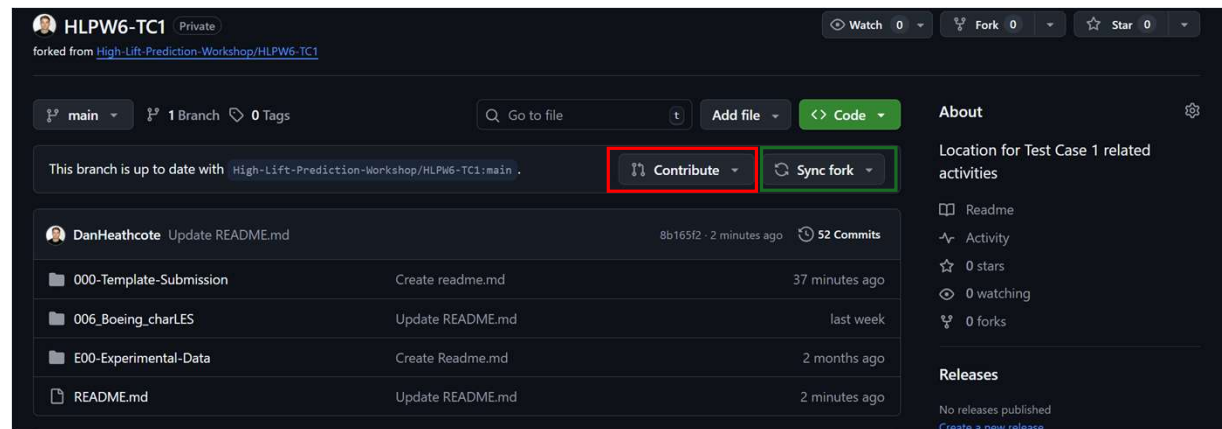
Copy the `main` branch only

Contribute back to High-Lift-Prediction-Workshop/HLPW6-TC1 by adding your own branch. [Learn more.](#)

You are creating a fork in your personal account.

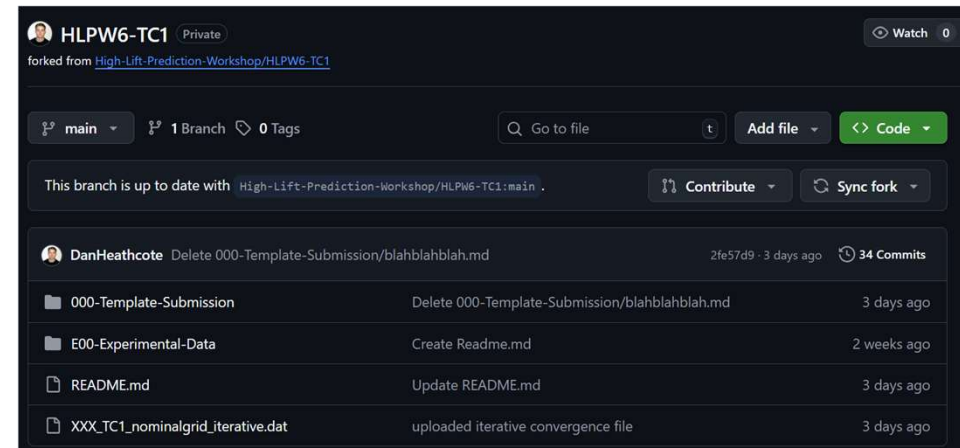
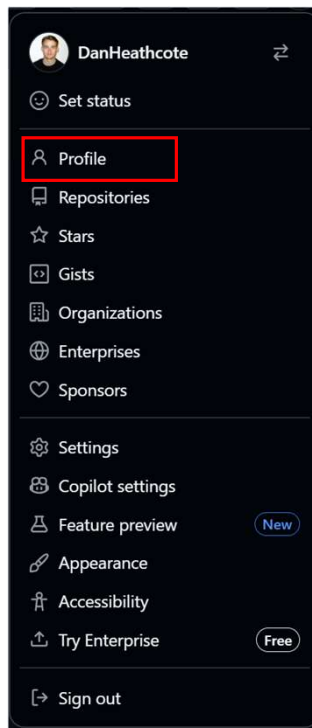
# Using your HLPW6-TC1 fork

- After creating the fork, you'll be brought to a landing page for your new fork of the repository.
- Some key features here are:
  - **Contribute** – This is how you will push (**submit**) your data up to the main repository
  - **Sync fork** – Clicking this will bring updates from the main repository into your local fork



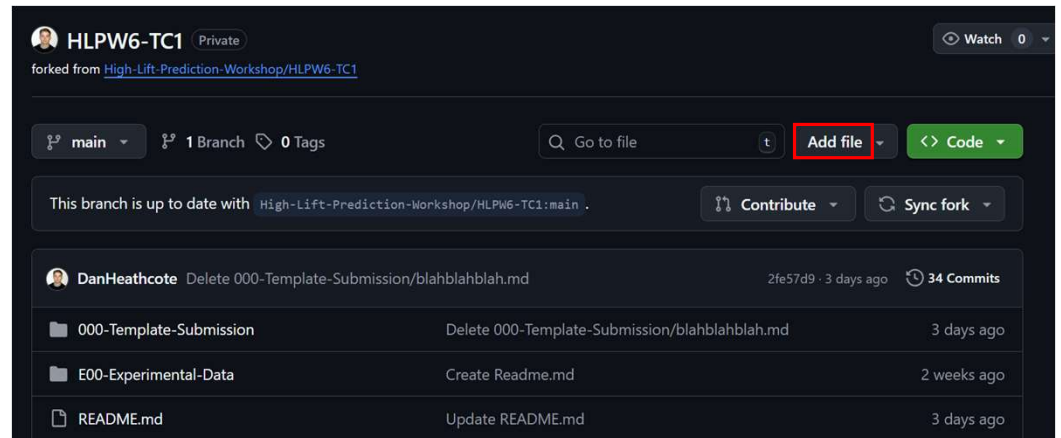
# Using your HLPW6-TC1 fork

- Ensure you are working within your fork:
  - Navigate to your profile by clicking on your profile image -> profile
  - Select you fork of **HLPW6-TC1**
- The view should match the right hand image (note **“forked from”** text)
- 000-Template-Submission contains the files which need to be modified for submission



# Creating a submission folder (1)

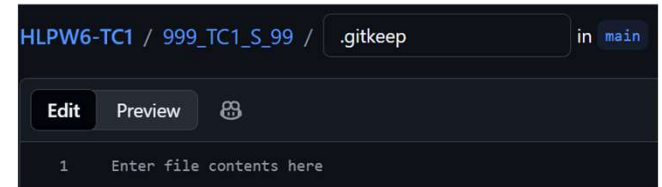
- Github does not natively allow browser based “copy” operations. The following provides a methodology to get around this. Alternate workflows via command line operations may be available
- Essentially, we need to copy the files from the template directory, modify them with our results, and add them to a folder corresponding to our participant ID, TFG and submission number
- Create a new folder by clicking **“add file”**



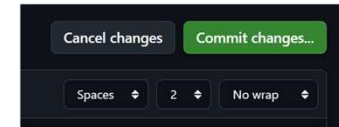
# Creating a submission folder (2)

- Folder naming convention should follow **HLPW6-TC1/readme.md**
- Input name in field after HLPW6-TC1 following convention adding **“/.gitkeep”** to the end
- As git tracks files and not folders, an empty file is required until the results files are uploaded
- Click **“Commit”** button on far right
- In pop up window, please provide a descriptive message for the commit
- Committing to main **branch** on your **fork** is appropriate

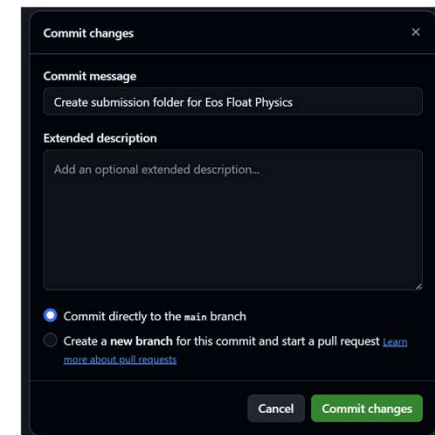
**Create**



**Commit**

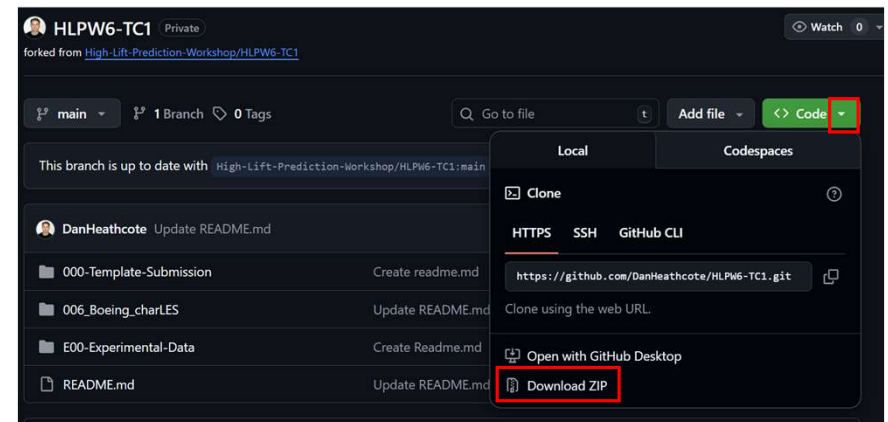
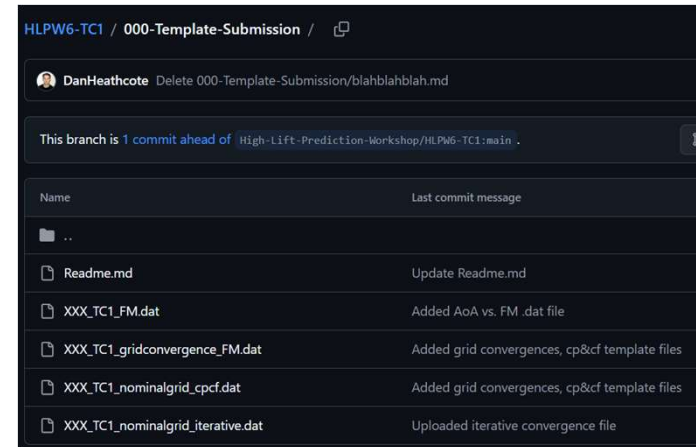


**Confirm**



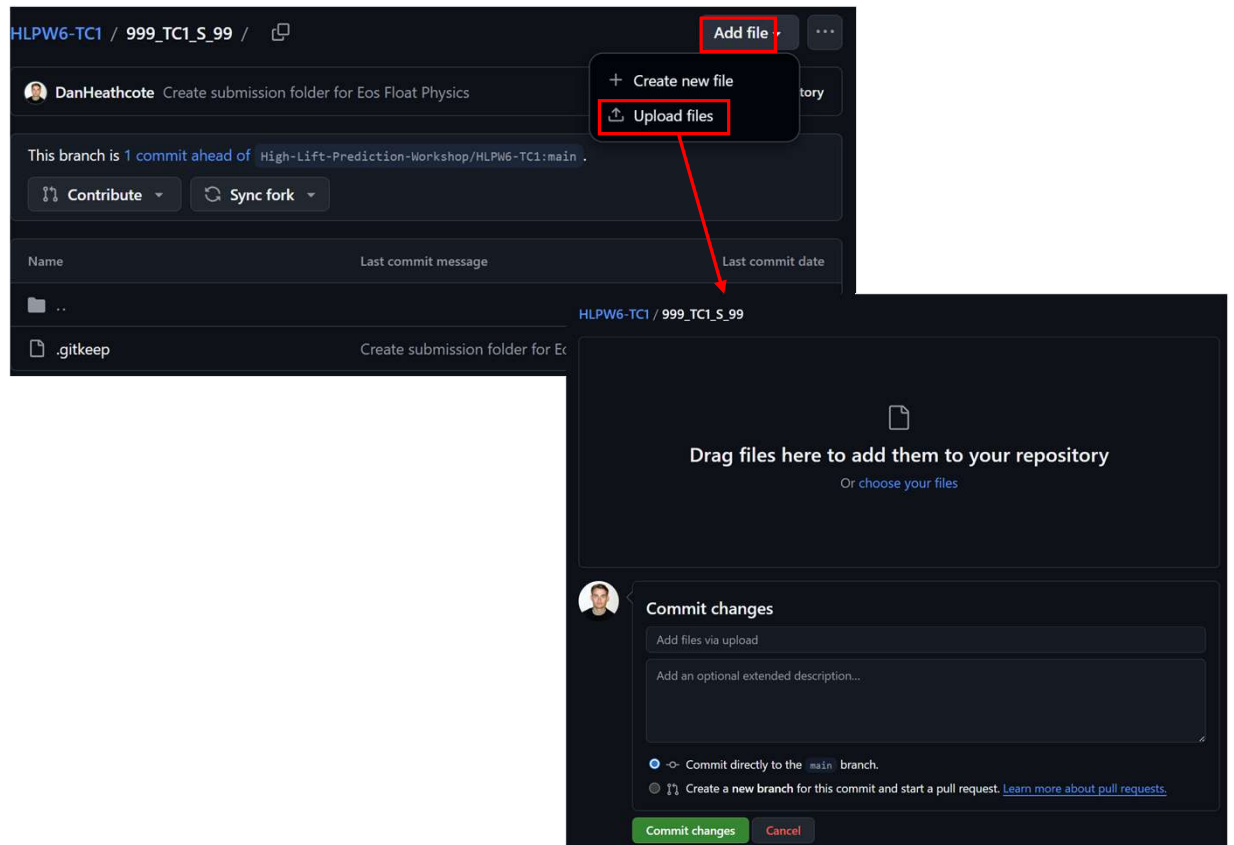
# Creating submission files

- Five files are provided in the template submission folder:
  1. **Readme.md** = contains supplemental information for the submission
  2. **FM.dat** = Converged or time-averaged Force and Moment data vs. angle of attack, multiple grid levels may be included
  3. **gridconvergence\_FM.dat** = Converged or time-averaged Force and Moment data vs. grid level, multiple angles of attack may be included (this is a transpose of #2)
  4. **nominalgrid\_cpcf.dat** = Surface pressure and skin friction distributions at locations corresponding to experimental pressure belts. These are defined here:
  5. **nominalgrid\_iterative.dat** = Solver iterative convergence with respect to iteration, or if time-dependent, with respect to convective time
- It's recommended to modify these by downloading a copy of the repo via a zip file (left)
- Alternatively, a local clone of the repo can be conducted



# Uploading Submission files (1/3)

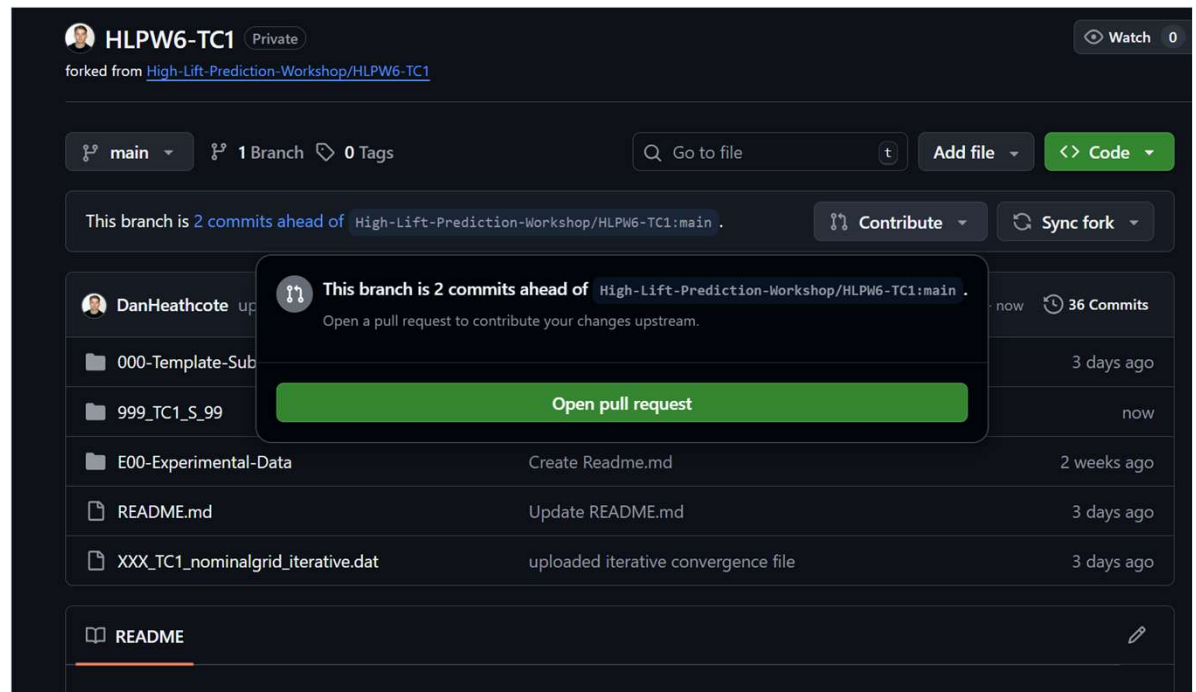
- Navigate back to your submission folder
- Click **“Add file”**, then **“Upload files”**
- Either drag & drop or navigate to submission file location
- Provide description to commit, and commit your changes



The screenshot shows the GitHub web interface for a repository named 'HLPW6-TC1 / 999\_TC1\_S\_99'. The user 'DanHeathcote' is shown with the message 'Create submission folder for Eos Float Physics'. The interface indicates 'This branch is 1 commit ahead of High-Lift-Prediction-Workshop/HLPW6-TC1:main'. There are buttons for 'Contribute' and 'Sync fork'. A dropdown menu is open under 'Add file', with 'Upload files' highlighted. A red arrow points from this menu to a modal window titled 'Commit changes'. The modal window has a header 'Drag files here to add them to your repository' with a sub-link 'Or choose your files'. Below this is a text input field for 'Add files via upload' and a larger text area for 'Add an optional extended description...'. At the bottom, there are radio buttons for 'Commit directly to the main branch.' (selected) and 'Create a new branch for this commit and start a pull request. Learn more about pull requests.'. There are 'Commit changes' and 'Cancel' buttons at the bottom of the modal.

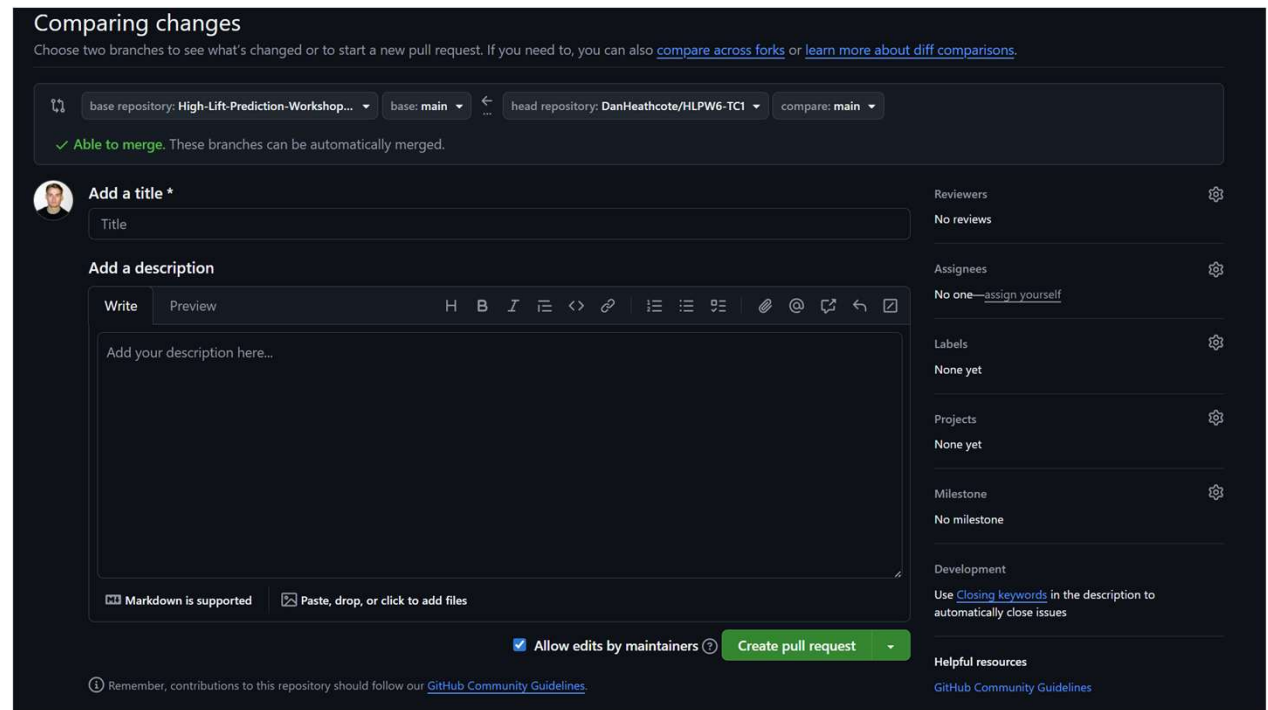
# Uploading Submission files (2/3)

- Your files are now uploaded to your fork of the repository.
- You can add more submissions by repeating the process of creating a submission folder and adding additional files.
- Once all of the files are uploaded to your fork, navigate to the front page of your repo, and click **“Contribute”** and **“Open pull request”**



# Uploading Submission files (3/3)

- From here, you'll have to add a title for your pull request, fill in some description, as necessary, and then click “**Create pull request**”
- This will inform the group leads that your data files are awaiting review
- Thank you for your contribution to HLPW6!
- The same fork can be used to create subsequent submittals by creating a new folder and completing the same “**pull request/push process**”



The screenshot shows the GitHub 'Comparing changes' interface. At the top, it says 'Comparing changes' and 'Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#) or [learn more about diff comparisons](#).' Below this, there are dropdown menus for 'base repository: High-Lift-Prediction-Workshop...', 'base: main', 'head repository: DanHeathcote/HLPW6-TC1', and 'compare: main'. A green checkmark indicates 'Able to merge. These branches can be automatically merged.'

The main form has two sections: 'Add a title \*' and 'Add a description'. The 'Add a title \*' section has a text input field with 'Title' as a placeholder. The 'Add a description' section has a rich text editor with a 'Write' tab selected and a 'Preview' tab. The text area contains the placeholder 'Add your description here...'. Below the text area, there are icons for 'Markdown is supported' and 'Paste, drop, or click to add files'.

On the right side, there are several sections: 'Reviewers' (No reviews), 'Assignees' (No one—assign yourself), 'Labels' (None yet), 'Projects' (None yet), 'Milestone' (No milestone), and 'Development' (Use [Closing keywords](#) in the description to automatically close issues). At the bottom right, there is a 'Helpful resources' section with a link to 'GitHub Community Guidelines'.

At the bottom of the form, there is a checkbox for 'Allow edits by maintainers' which is checked, and a green 'Create pull request' button.

# Final thoughts (from DPW)

- This method only relied on the web interface for GitHub. If you're familiar with their command line interface or GitHub application, you absolutely can use those as well.
- Undoubtedly people will run into issues along the way, so please feel free to reach out to the group leads with any questions.
- Also, there is a wealth of information on the Fork and Pull method of contributing to a repository. Here's one tutorial that is slightly more advanced and uses the command line interface:  
<https://gist.github.com/Chaser324/ce0505fbed06b947d962>