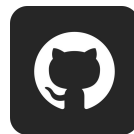


How to build a private customized peft package

TAs: 黃筱穎、陳又華、謝翔

Email: ntu-ml-2025-spring-ta@googlegroups.com



Generate a private repository from a public github repo

- Generate a **private** repository from TA-version peft github repo (Use this template)

<https://github.com/chenjoachim/peft-ml2025-hw9.git>

Generate a private repository from a public github repo

chenjoachim / peft-ml2025-hw9

Search Type / to search

+

+

+

+

+

+

<> Code

Issues

Pull requests

Actions

Projects

Security

Insights

peft-ml2025-hw9

Public template

Watch 1

Fork 0

Star 0

Use this template

Create a new repository

Open in a codespace

main

1 Branch

0 Tags

Go to file

Add file

<> Code

About

chenjoachim Disable GitHub Actions

9027c53 · 28 minutes ago

4 Commits

.github/ISSUE_TEMPLATE	Disable GitHub Actions	28 minutes ago
docker	Initial commit	39 minutes ago
docs	Initial commit	39 minutes ago
examples	Initial commit	39 minutes ago
scripts	Initial commit	39 minutes ago
src/peft	Initial commit	39 minutes ago

No description, website, or topics provided.

Readme

Apache-2.0 license

Activity

0 stars

1 watching

0 forks

Report repository

Releases



New repository

🔍 Type to search



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?

[Import a repository.](#)

Required fields are marked with an asterisk ().*

Repository template



chenjoachim/peft-ml2025-hw9 ▾

Start your repository with a template repository's contents.



Include all branches

Copy all branches from chenjoachim/peft-ml2025-hw9 and not just the default branch.

Owner *



monica0325 ▾



Repository name *


peft_private



✔ peft_private is available.

Great repository names are short and memorable. Need inspiration? How about [vigilant-lamp](#) ?

Owner *

 monica0325 ▾

Repository name *

/ peft_private

✓ peft_private is available.

Great repository names are short and memorable. Need inspiration? How about **vigilant-lamp** ?

Description (optional)



Public

Anyone on the internet can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.



You are creating a private repository in your personal account.

Create repository

Clone the private github repository using the web url

The screenshot shows the GitHub interface for a repository named **peft_private**, which is marked as **Private**. The repository was generated from [chenjoachim/peft-ml2025-hw9](#). At the top, there are buttons for **Unwatch** (1), **Fork** (0), and **Star** (0). Below the repository name, there are tabs for **main**, **1 Branch**, and **0 Tags**. A search bar labeled "Go to file" and buttons for **Add file** and **<> Code** are also visible. The **Code** button is highlighted with a red circle, and its dropdown menu is open, showing the **Clone** option selected. The **Clone** menu has two tabs: **Local** and **Codespaces**. Under the **Local** tab, there are three options: **HTTPS** (selected), **SSH**, and **GitHub CLI**. The **HTTPS** option shows the URL `https://github.com/monica0325/peft_private.git` with a copy icon. Below the URL, it says "Clone using the web URL." There are also options to **Open with GitHub Desktop** and **Download ZIP**. On the right side of the repository page, there is an **About** section with a gear icon, stating "No description, website, or topics provided." Below this, there are links for **Readme**, **Apache-2.0 license**, **Activity**, **0 stars**, **1 watching**, and **0 forks**. At the bottom, there is a **Releases** section with the text "No releases published".

peft_private Private
generated from [chenjoachim/peft-ml2025-hw9](#)

main 1 Branch 0 Tags

Go to file Add file <> Code

monica0325 Initial commit

- .github/ISSUE_TEMPLATE Initial commit
- docker Initial commit
- docs Initial commit
- examples Initial commit
- scripts Initial commit
- src/peft Initial commit

Local **Codespaces**

Clone ?

HTTPS **SSH** **GitHub CLI**

`https://github.com/monica0325/peft_private.git` Copy

Clone using the web URL.

Open with GitHub Desktop

Download ZIP

About ⚙️

No description, website, or topics provided.

- Readme
- Apache-2.0 license
- Activity
- 0 stars
- 1 watching
- 0 forks

Releases

No releases published

Clone a github repository using the web url

- Open your local terminal with [Git](#) installed and clone your new private github repo, for example:

```
git clone https://github.com/{username}/peft\_private.git
```

(replace with the url link of your private repo)

```
cd peft_private/
```

(cd the main directory)

[Git basic usage](#)

Clone the private github repository using the web url

```
user@ [REDACTED]
$ git clone https://github.com/monica0325/peft_private.git
Cloning into 'peft_private'...
remote: Enumerating objects: 405, done.
remote: Counting objects: 100% (405/405), done.
remote: Compressing objects: 100% (288/288), done.
remote: Total 405 (delta 105), reused 403 (delta 105), pack-reused 0 (from 0)
Receiving objects: 100% (405/405), 5.63 MiB | 483.00 KiB/s, done.
Resolving deltas: 100% (105/105), done.

user@ [REDACTED]
$ cd peft_private/

user@ [REDACTED] peft_private (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```


Git workflow to push local modifications to the remote repo

General **Git workflow**:

```
# Step 1. Check current status
git status

# Step 2. Stage your modified files or Stage all changes
git add filename1 filename2 # or git add .

# Step 3. Commit your changes
git commit -m "Describe what you've changed"

# Step 4. Pull latest remote changes (if collaborating)
git pull origin main

# Step 5. Push your committed changes to remote
git push origin main
```

```
user@ /peft_private (main)
● $ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   src/peft/utils/merge_utils.py

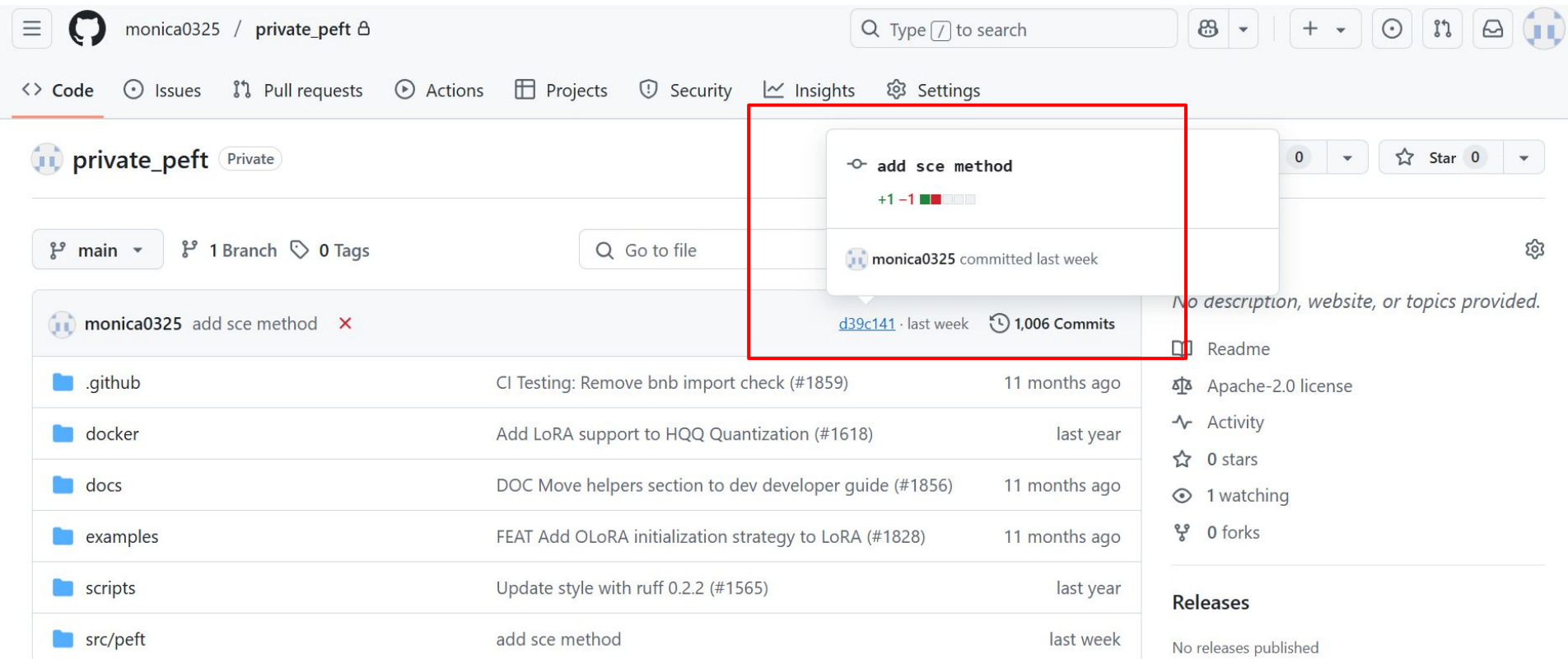
no changes added to commit (use "git add" and/or "git commit -a")

user@ /peft_private (main)
● $ git add src/peft/utils/merge_utils.py

user@ /peft_private (main)
● $ git commit -m "add sce in merge_utils"
[main c69f4b0] add sce in merge_utils
1 file changed, 21 insertions(+), 4 deletions(-)

user@ /peft_private (main)
● $ git push
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 16 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 804 bytes | 804.00 KiB/s, done.
Total 6 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
To https://github.com/monica0325/peft_private.git
bc89b2f..c69f4b0  main -> main
```

Derive a private customized peft package



monica0325 / private_peft

Type / to search

<> Code Issues Pull requests Actions Projects Security Insights Settings

private_peft Private

main 1 Branch 0 Tags

Go to file

monica0325 add sce method

add sce method

+1 -1

monica0325 committed last week

d39c141 · last week 1,006 Commits

No description, website, or topics provided.

Readme

Apache-2.0 license

Activity

0 stars

1 watching

0 forks

Releases

No releases published

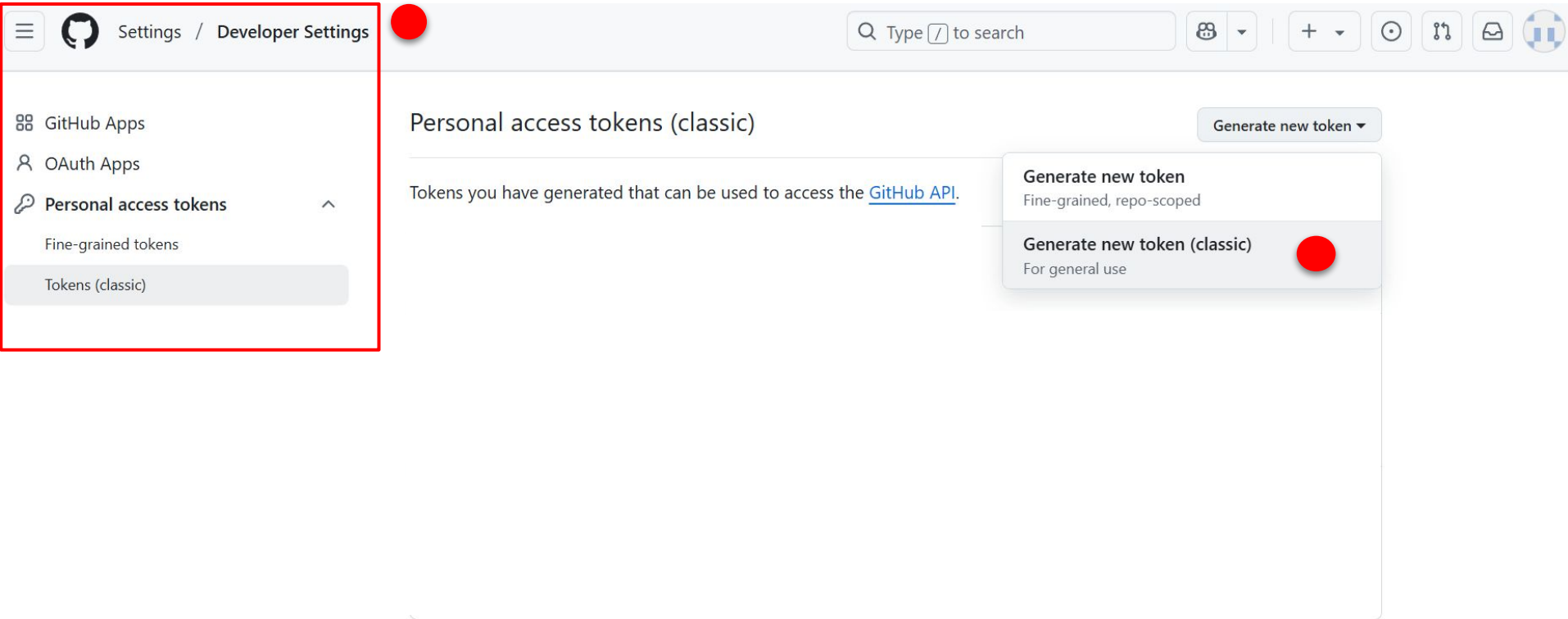
.github	CI Testing: Remove bnb import check (#1859)	11 months ago
docker	Add LoRA support to HQQ Quantization (#1618)	last year
docs	DOC Move helpers section to dev developer guide (#1856)	11 months ago
examples	FEAT Add OLoRA initialization strategy to LoRA (#1828)	11 months ago
scripts	Update style with ruff 0.2.2 (#1565)	last year
src/peft	add sce method	last week

Generate private repo access token on GitHub

- Generate a github repo **access token** for accessing/installing your **private repository** on Colab/Kaggle notebook

```
https://github.com/{username}/peft_private.git
```

Generate private repo access token on GitHub



The screenshot shows the GitHub Developer Settings page. On the left, a sidebar contains navigation links: GitHub Apps, OAuth Apps, and Personal access tokens. The 'Personal access tokens' link is highlighted with a red box. Below it, there are two sub-options: 'Fine-grained tokens' and 'Tokens (classic)', with 'Tokens (classic)' being the selected option. The main content area is titled 'Personal access tokens (classic)' and includes a 'Generate new token' button. Below this, a text line states 'Tokens you have generated that can be used to access the [GitHub API](#).' A dropdown menu is open, showing two options: 'Generate new token' (Fine-grained, repo-scoped) and 'Generate new token (classic)' (For general use). The 'Generate new token (classic)' option is highlighted with a red circle.

Settings / Developer Settings

Search Type / to search

Personal access tokens (classic)

Generate new token

Tokens you have generated that can be used to access the [GitHub API](#).

Generate new token
Fine-grained, repo-scoped

Generate new token (classic)
For general use

GitHub Apps

OAuth Apps

Personal access tokens

Fine-grained tokens

Tokens (classic)

New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Note

private repo access

What's this token for?

Expiration

30 days (Jun 10, 2025)

The token will expire on the selected date

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes](#).

- | | |
|---|--|
| <input checked="" type="checkbox"/> repo | Full control of private repositories |
| <input checked="" type="checkbox"/> repo:status | Access commit status |
| <input checked="" type="checkbox"/> repo_deployment | Access deployment status |
| <input checked="" type="checkbox"/> public_repo | Access public repositories |
| <input checked="" type="checkbox"/> repo:invite | Access repository invitations |
| <input checked="" type="checkbox"/> security_events | Read and write security events |
| <input type="checkbox"/> workflow | Update GitHub Action workflows |
| <input type="checkbox"/> write:packages | Upload packages to GitHub Package Registry |
| <input checked="" type="checkbox"/> read:packages | Download packages from GitHub Package Registry |

Install a private GitHub repository as a pip package on Colab or Kaggle

```
# Step 1: Set GitHub username and repo name
username = "your_github_username"
repo_name = "your_private_repo_name"

# Step 2: Enter GitHub token securely
token = getpass("Enter your GitHub token (it will be hidden): ")
os.environ["GITHUB_TOKEN"] = token # optional: store in env for reuse
# or token = "github_repo_access_token"

# Step 3: Construct repo URL and install via pip
repo_url = f"git+https://{username}:{token}@github.com/{username}/{repo_name}.git"

# Step 4: Install
!pip install {repo_url}
```



End of the Tutorial! Thank you~