

# PyTorch's Top 5

Eli Uriegas (@seemethere)

PyTorch @Meta

RIGHT CLICK >  
CHANGE PICTURE

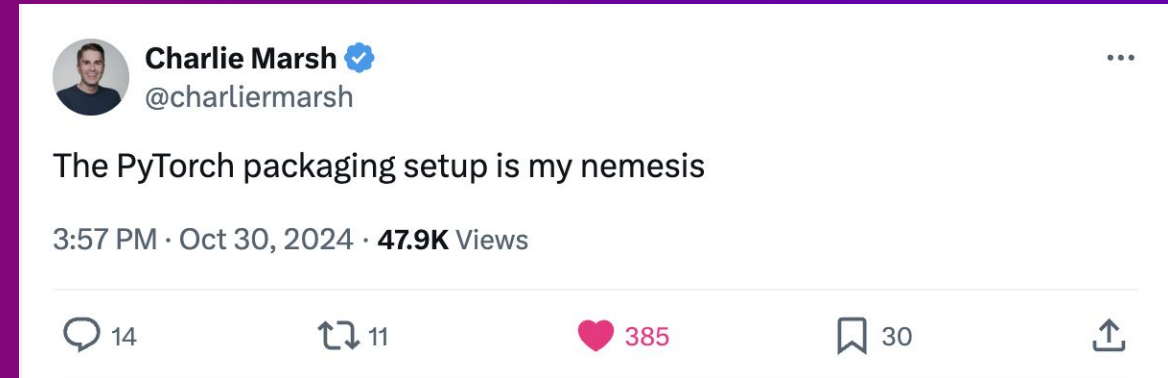
# What are trying to build?

- PyTorch is trying to build the best foundation for ML based applications
  - As well we're trying to inspire community amongst a large variety of industry leaders to work together on this goal



# What are challenges / blockers?

- PyPI space limits regularly haunt us
  - Hard to get space limit increases
  - Hinders our ability to split up the Pytorch package into smaller subpackages
- PyTorch needs to host its own package index to support multiple hardware accelerators
  - Leads to churn on install instructions / how different package managers handle PyTorch
  - Security conflicts with public PyTorch (see PEP 708)
- Improvements to packaging held back by lack of features like default-extras



```
# Based on accelerator type
pip install torch[cpu]
pip install torch[cuda]
pip install torch[rocm]
pip install torch[xpu]

# Based on accelerator version
pip install torch[cuda12-4]
pip install torch[cuda12-1]
pip install torch[rocm6-2]
pip install torch[rocm6-1]
```

# What are you looking ahead to improve/adopt?

- variants look great!
  - We want to help remove the local identifier experience from PyTorch's packaging
- Index prioritization is also a great one too!
  - Would help with users utilizing multiple indices
- Default extra would make transitioning the experience of using *pip install torch* a lot better until variants are up!
- symlinks in wheels would be awesome
  - The less duplication the better

### 3 things that work well / could be improved

- manylinux has been a great standard to build for
  - Removes ambiguity over which ABI we support
- build-system (PEP 518) is cool but incomplete (from our perspective)
  - Would like more features here around specifying particular compilers to use (nvcc, rocmcc, etc.)
  - conda-like system to specify toolchain would be great here
- Handling of multiple indices is on track to get better
  - Need to finish out PEP 708 / PEP 766

## How we'd like to contribute

- Contribute to variants PEP
- Contribute to PEPs supporting multiple indices (708, 766)
- Contribute to overall organization of WheelNext Initiative