# Adopting static typing in scientific projects

Predrag Gruevski & Colin Carroll SciPy 2021

# **Speakers**



Predrag Gruevski

typing\_copilot author

Twitter: @PredragGruevski



Colin Carroll

ArviZ maintainer

Twitter: @colindcarroll

# Python static typing and type hints

\_\_\_\_

```
def square(a):
return a * a
```

```
def square(a: int) -> int:
    return a * a
```

# **Outline**

- Difficulties when type-hinting a large existing codebase
- "Ratcheting" type-hinting workflow with typing\_copilot
- Case studies on adopting typing\_copilot

# **Outline**

- Difficulties when type-hinting a large existing codebase
- "Ratcheting" type-hinting workflow with typing\_copilot
- Case studies on adopting typing\_copilot

### **Three Stakeholder Personas**

\_\_\_\_

End-user Emily

C Contributor Cathy



# Type-hinting a large existing codebase

End-user Emily

#### Wants new features!

- Type hints and features must be added concurrently
- Wants type hints exported to support her own codebase

# Type-hinting a large existing codebase

Contributor Cathy

Might not be familiar w/ type-hinting. Might stop contributing if:

- Adding type hints is too difficult or too time-consuming
- Pull request reviews take too long

# Type-hinting a large existing codebase

Maintainer Michelle

Needs to keep 

and 

happy.

#### Also wants:

- Type hints used to prevent bugs
- Fully type-hinted new features
- Existing code allowed to be partially type-hinted
- Type hint coverage cannot degrade

# Difficulties when adopting type-hinting

#### Difficult up-front setup for M



- Different rules for different parts of the codebase
- Complex mypy config required
- Lots of config options, which are best?
- => Analysis paralysis! M might give up entirely...

# Difficulties when adopting type-hinting

Difficult up-front setup for M



- Different rules for different parts of the codebase
- Complex mypy config required
- Lots of config options, which are best?
- => Analysis paralysis! M might give up entirely...

# Ongoing difficulties for (C) and (M)

- Must ensure pull requests do not degrade type hint coverage
- Must update mypy config as type hint coverage improves
- => Two steps forward, one step back

# Difficulties when adopting type-hinting

#### Difficult up-front setup for M



- Different rules for different parts of the codebase
- Complex mypy config required
- Lots of config options, which are best?
- => Analysis paralysis! M might give up entirely...

# Ongoing difficulties for (C) and (M)

- Must ensure pull requests do not degrade type hint coverage
- Must update mypy config as type hint coverage improves

#### => Two steps forward, one step back

# **Outline**

- Difficulties when type-hinting a large existing codebase
- "Ratcheting" type-hinting workflow with typing\_copilot
- Case studies on adopting typing\_copilot

typing\_copilot

#### Finds the strictest mypy config that passes with no errors

- Checks codebase with all mypy checks enabled
- Any checks that find errors are disabled for the affected files
- Everything else represents the strictest passing mypy config!

# typing\_copilot resolves type-hinting difficulties

No analysis paralysis

- The best mypy config is always the strictest passing one
- typing\_copilot init

# typing\_copilot resolves type-hinting difficulties

# No analysis paralysis

- The best mypy config is always the strictest passing one
- typing\_copilot init

## Ratcheting workflow (C) (M)





- Catch type correctness or type coverage regressions with mypy
- Then, try to generate a stricter passing mypy config and check it in
- On CI, ensure the mypy config is the strictest possible
- typing\_copilot tighten [--error-if-can-tighten]

# **Outline**

- Difficulties when type-hinting a large existing codebase
- "Ratcheting" type-hinting workflow with typing\_copilot
- Case studies on adopting typing\_copilot

# Actionable and minimal path to type-checking ArviZ

Running mypy arviz produced 361 type errors!

- Most mypy checks are enabled by default
- Fixing hundreds of errors is not an encouraging starting point (C)





# Actionable and minimal path to type-checking ArviZ

Running mypy arviz produced 361 type errors!

- Most mypy checks are enabled by default
- Fixing hundreds of errors is not an encouraging starting point





#### typing\_copilot init produces the minimal set of errors

- Runs mypy with all optional checks disabled
- Any failures in that configuration must be fixed before proceeding
- ArviZ had 41 type errors with all optional checks disabled
- Multiple user-facing bugs fixed in the process, e.g.: <a href="https://github.com/arviz-devs/arviz/pull/1493">https://github.com/arviz-devs/arviz/pull/1493</a>





# Sustainably improving type coverage in GraphQL compiler

\_\_\_

typing\_copilot requires this module to be completely typed

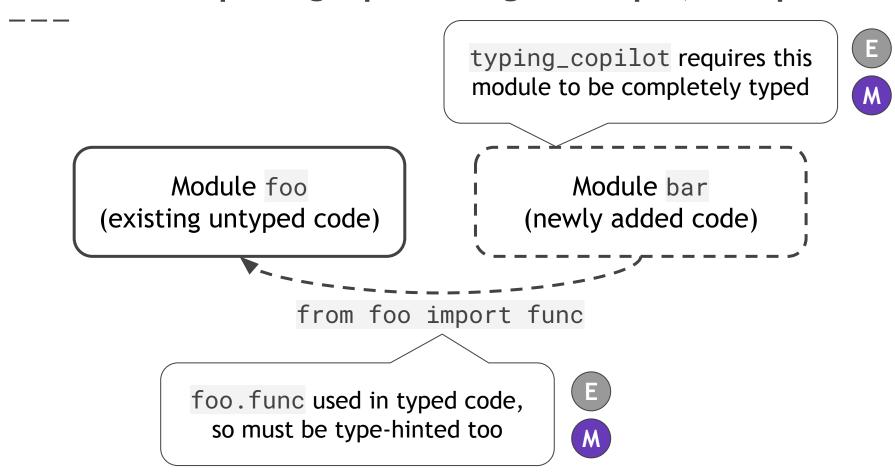
E



Module foo (existing untyped code)

Module bar (newly added code)

# Sustainably improving type coverage in GraphQL compiler



Mypy's disallow\_untyped\_calls rule:

"don't allow calling untyped functions from type-hinted code."

# Bonus ratchet tightening on files you didn't modify

This PR didn't modify the file referenced by this rule! 14 mypy.ini 79 [mypy-graphql\_compiler.compiler.helpers.\*] 80 - disallow untyped calls = False Comment on lines -79 to -80 obi1kenobi on Aug 6, 2020 Author Member This is perhaps my favorite part about typing-copilot — it catches and tightens settings I never would have thought about doing by hand. In this case, it noticed that graphql compiler.compiler.helpers was calling a previously-untyped function that is now typed. Since this suppression is no longer necessary, it is removed. That module is guaranteed to now and forever more only call fully-typed code.

https://github.com/kensho-technologies/graphql-compiler/pull/891/files#r466729797

# Try typing\_copilot in your own project

\_\_\_\_

pip install typing\_copilot

Repo: <a href="https://github.com/obi1kenobi/typing\_copilot">https://github.com/obi1kenobi/typing\_copilot</a>

Examples of adopting typing\_copilot + plugging into CI:

https://github.com/kensho-technologies/graphql-compiler/pull/937

https://github.com/arviz-devs/arviz/pull/1528

https://github.com/arviz-devs/arviz/pull/1624

Talk to us on Twitter:

@PredragGruevski and @colindcarroll