

Copilot

You can modify your setter to accept either a dictionary or a tuple. If it's a tuple, you can consider it as a key-value pair and update only that key in the dictionary. Here's how you can do it:

```
class MyClass:
    def __init__(self, params: dict = {"T": 1.0, "D": 0.5}) -> None:
        self._params: dict = params

    @property
    def params(self) -> dict:
        """ Get or set the parameters """
        return self._params

    @params.setter
    def params(self, params) -> None:
        if isinstance(params, dict):
            self._params = params
        elif isinstance(params, tuple) and len(params) == 2:
            key, value = params
            self._params[key] = value
        else:
            raise ValueError("params must be a dict or a tuple of length 2")
```

Now, you can update a single key-value pair like this:

```
obj = MyClass()
obj.params = ("T", 2.0) # This will update only the 'T' key in the params dictionary
```

This way, you can update the entire dictionary or just a single key-value pair using the same property. Remember to handle the case where the input is not a dictionary or a 2-tuple to prevent errors.