

single_vs_double_precision_meanprofile

June 6, 2024

```
[1]: import xarray as xr
import numpy as np
import matplotlib.pyplot as plt
import sys
```

```
[2]: output_dir1 = './output32'
output_dir2 = './output64'
case_output = '/output_twice_SCM_RAP'

f32 = output_dir1 + case_output + '/output.nc'
f64 = output_dir2 + case_output + '/output.nc'

ds32 = xr.open_dataset(f32)
ds64 = xr.open_dataset(f64)
```

```
[3]: ds32.identical(ds64)
```

[3]: False

```
[4]: print("single precision var is red, double precision var is blue")
# compare variables across full run
for var in ds32.variables:
    if (not ds32[var].identical(ds64[var])):
        if ('vert_dim_layer' in ds32[var].dims):
            fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(12, 4))
            ds32[var].mean(dim="vert_dim_layer").plot(ax=axes[0],c='r')
            ds64[var].mean(dim="vert_dim_layer").plot(ax=axes[0],c='b')
            axes[0].set_title(var+' mean profile')
            (ds64[var].mean(dim="vert_dim_layer")-ds32[var].
↳mean(dim="vert_dim_layer")).plot(ax=axes[1])

            axes[1].set_title(var+' profile difference')
            plt.show()
        elif ('vert_dim_interface' in ds32[var].dims):
            fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(12, 4))
            ds32[var].mean(dim="vert_dim_interface").plot(ax=axes[0],c='r')
            ds64[var].mean(dim="vert_dim_interface").plot(ax=axes[0],c='b')
            axes[0].set_title(var+' mean profile')
```

```

(ds64[var].mean(dim="vert_dim_interface")-ds32[var].
↳mean(dim="vert_dim_interface")).plot(ax=axes[1])

axes[1].set_title(var+' profile difference')
plt.show()
elif ('vert_dim_rad' in ds32[var].dims):
fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(12, 4))
ds32[var].mean(dim="vert_dim_rad").plot(ax=axes[0],c='r')
ds64[var].mean(dim="vert_dim_rad").plot(ax=axes[0],c='b')
axes[0].set_title(var+' mean profile')
(ds64[var].mean(dim="vert_dim_rad")-ds32[var].
↳mean(dim="vert_dim_rad")).plot(ax=axes[1])

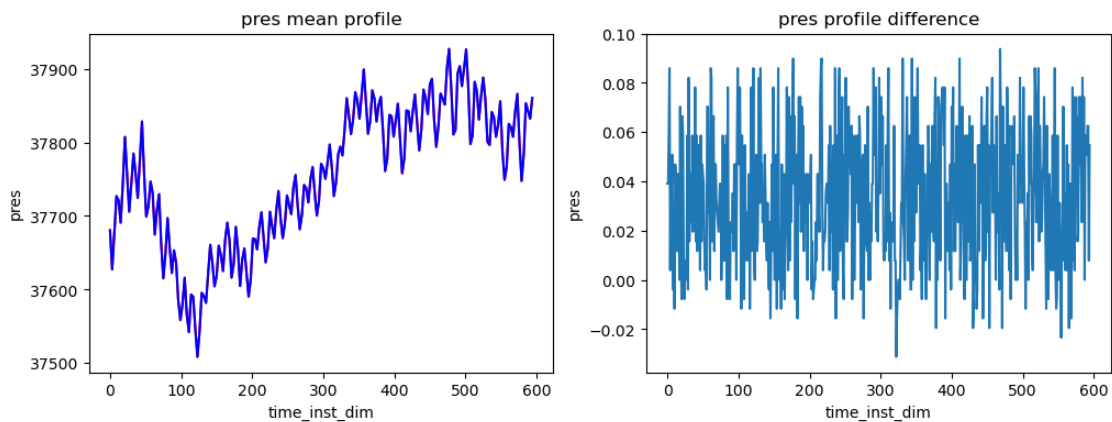
axes[1].set_title(var+' profile difference')
plt.show()
else:
fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(12, 4))
ds32[var].plot(ax=axes[0],c='r')
ds64[var].plot(ax=axes[0],c='b')
axes[0].set_title(var+' mean profile')
(ds64[var]-ds32[var]).plot(ax=axes[1])

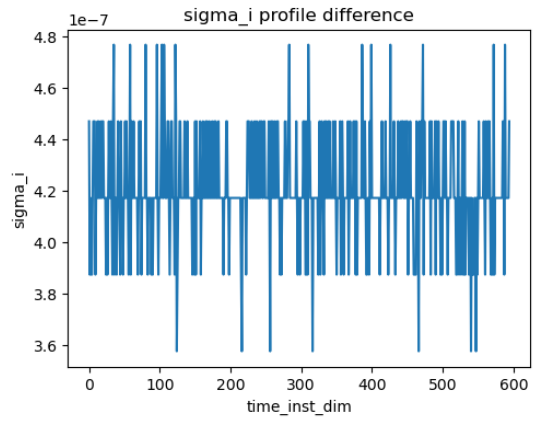
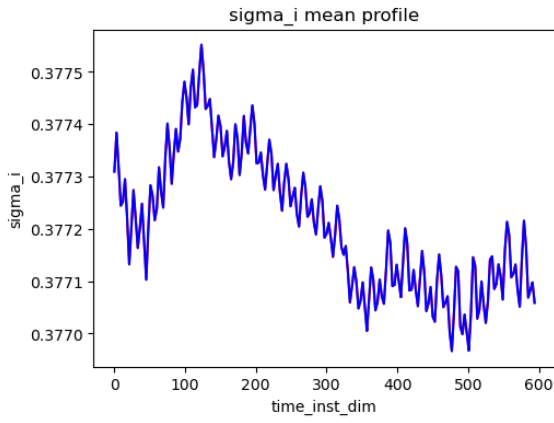
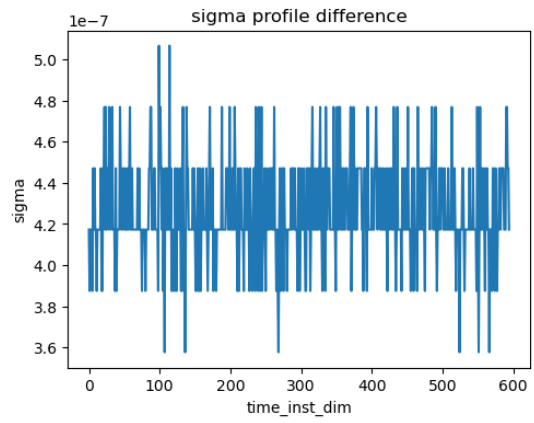
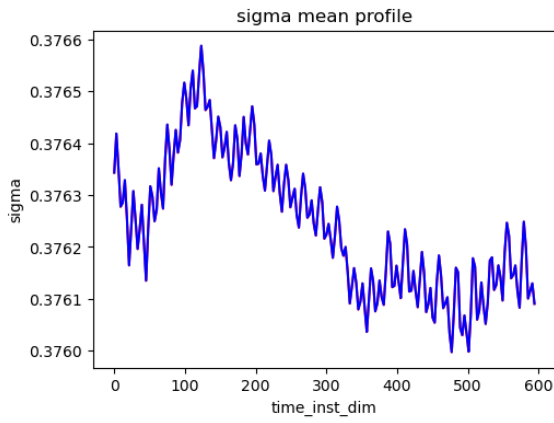
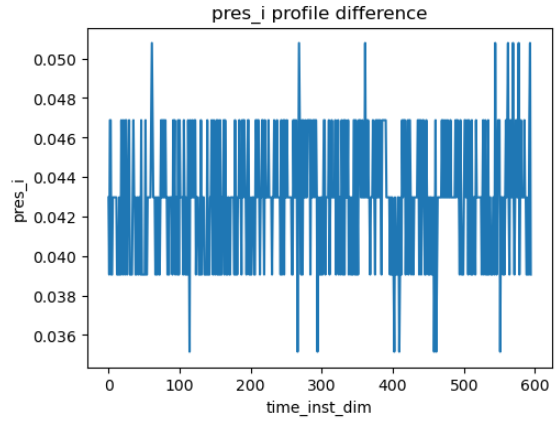
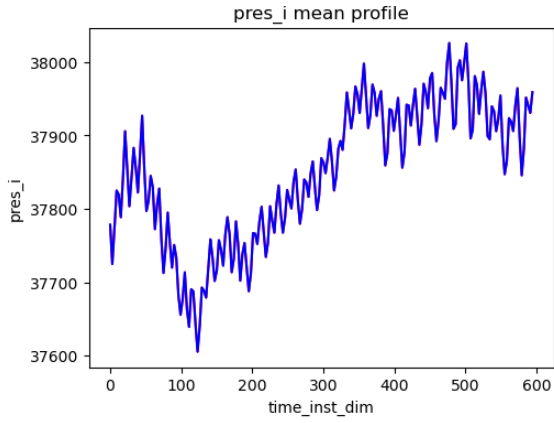
axes[1].set_title(var+' profile difference')

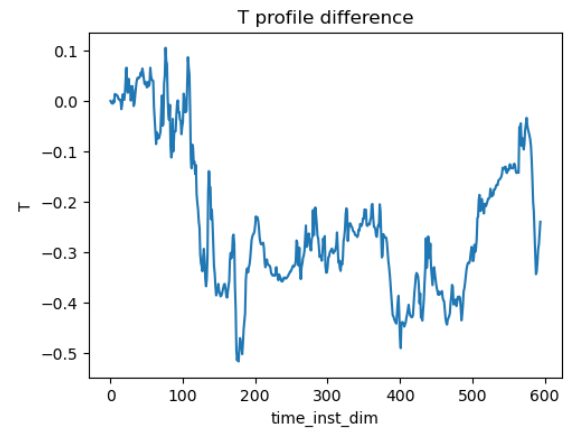
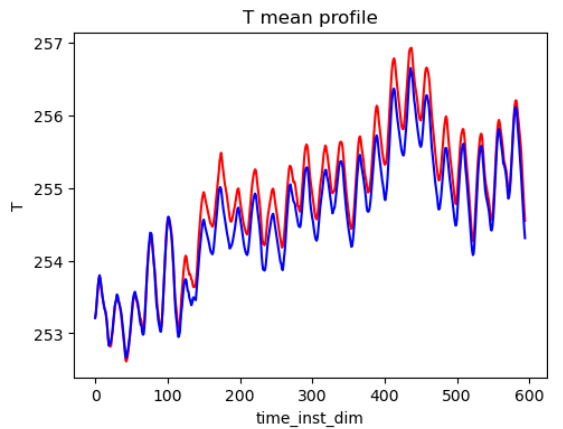
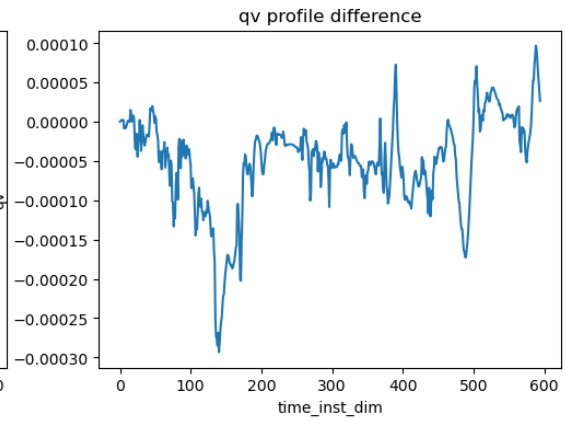
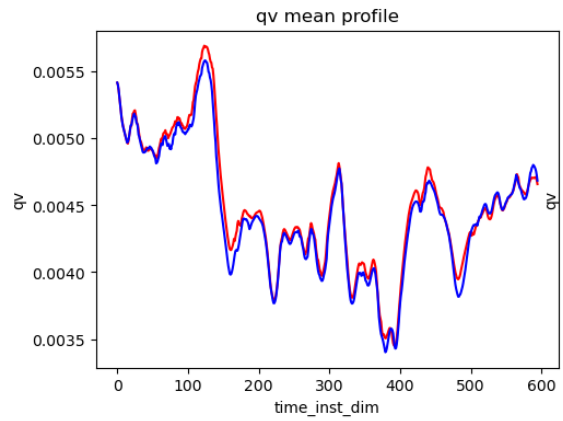
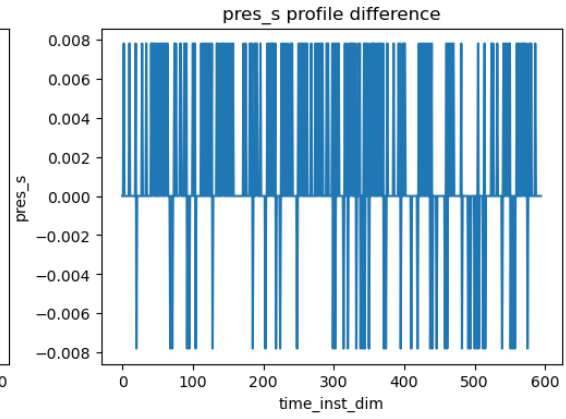
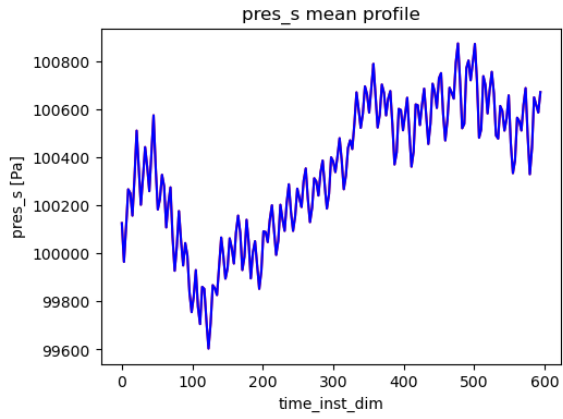
plt.show()

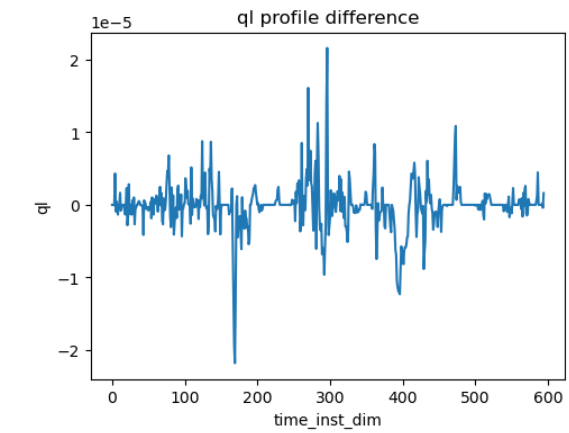
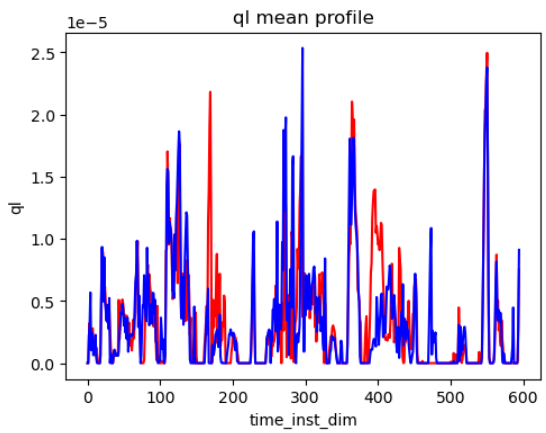
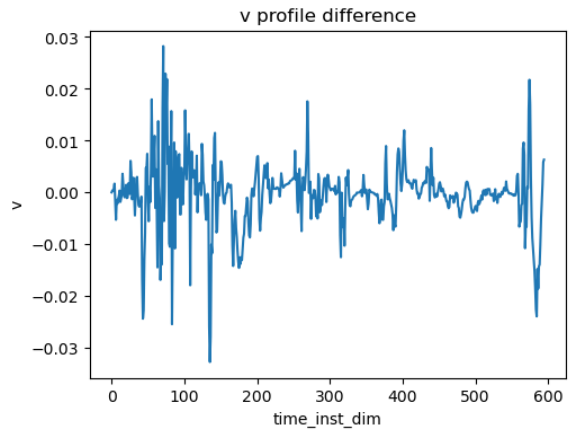
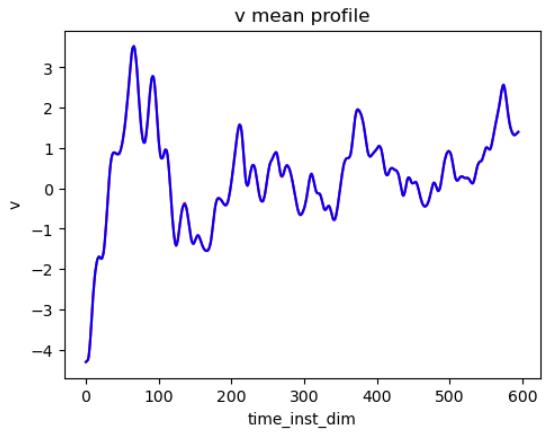
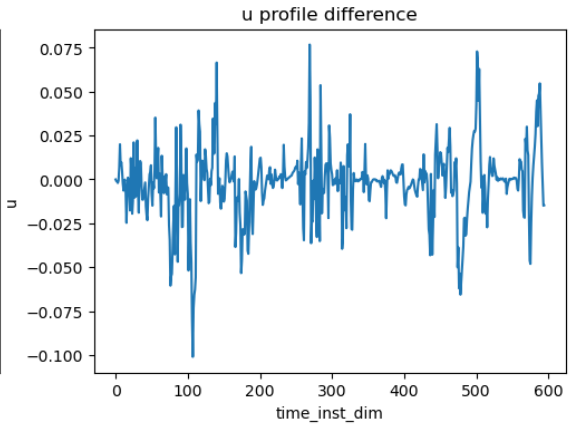
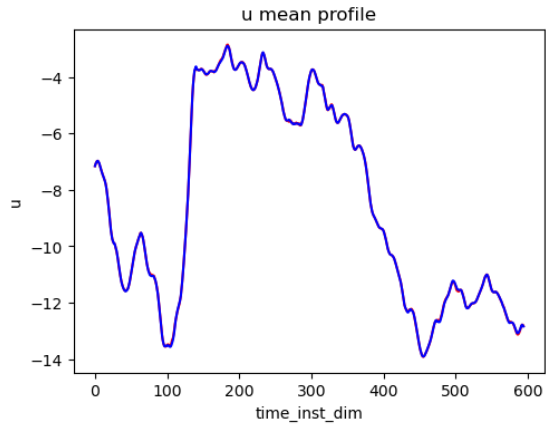
```

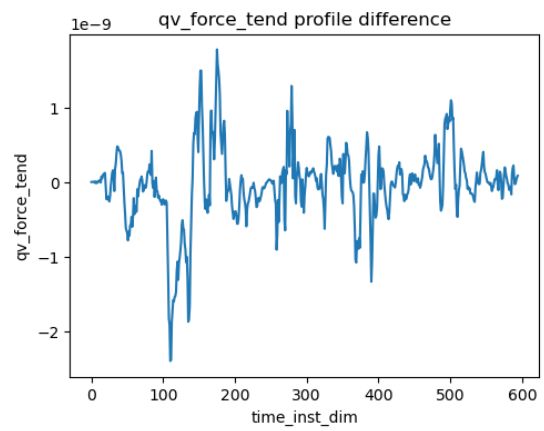
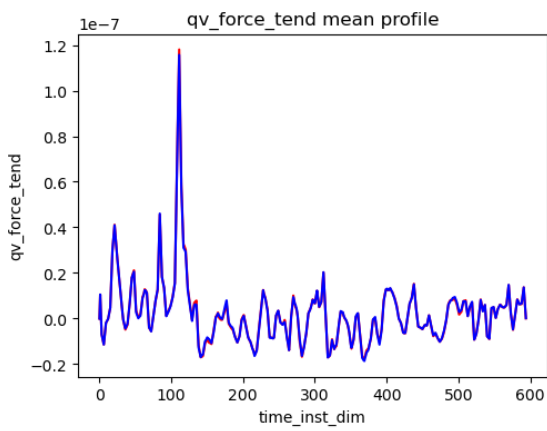
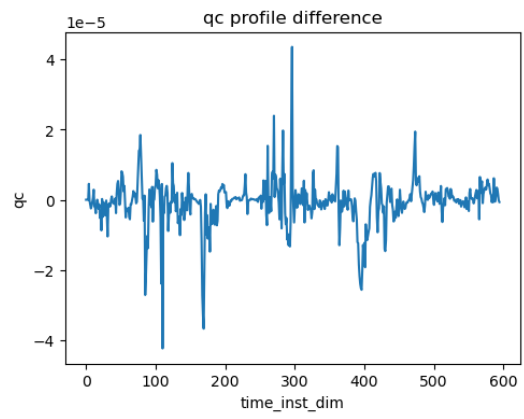
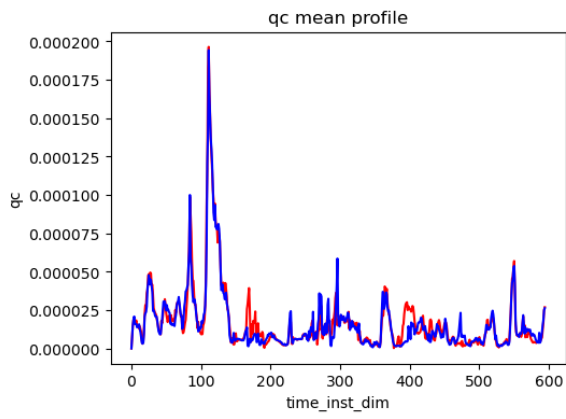
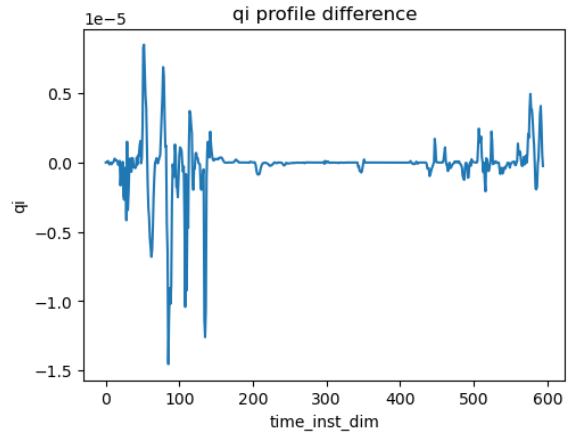
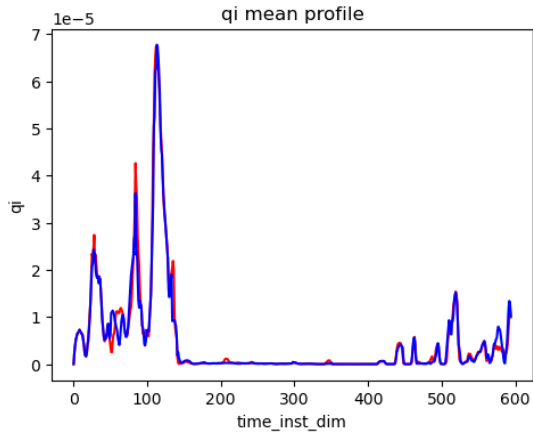
single precision var is red, double precision var is blue

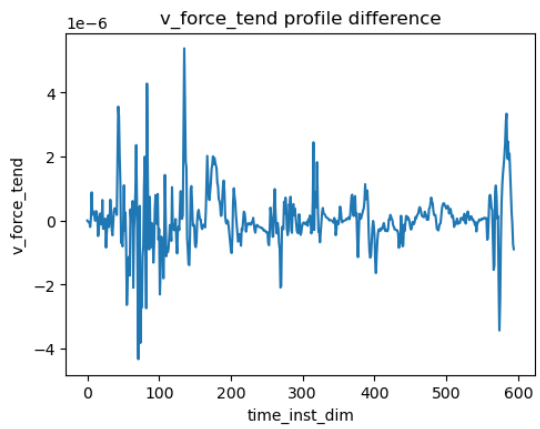
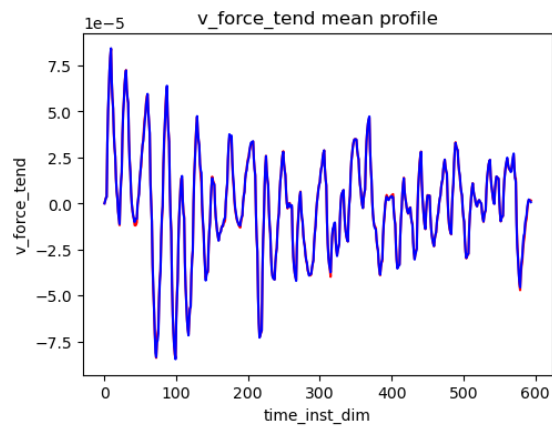
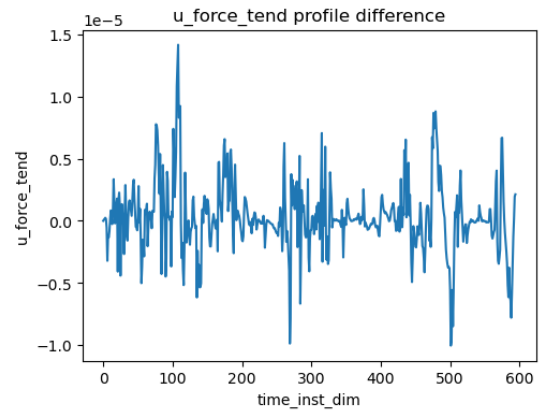
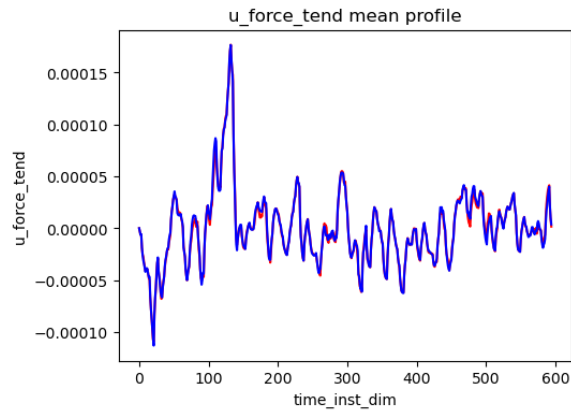
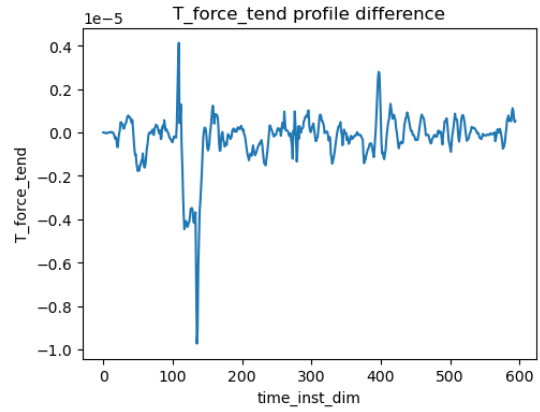
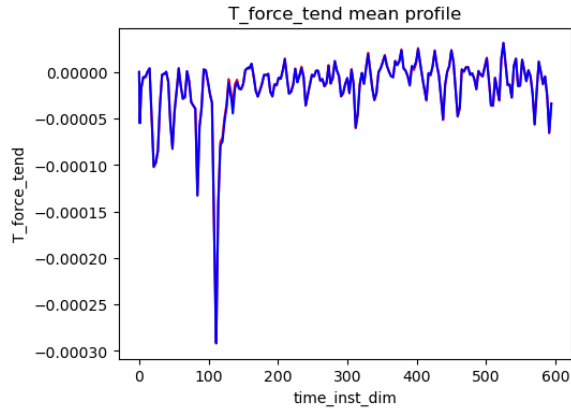


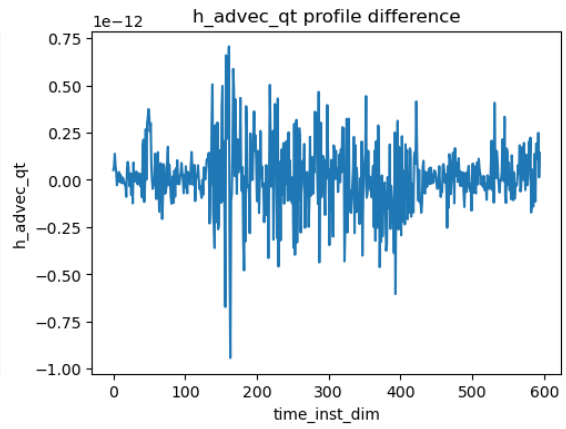
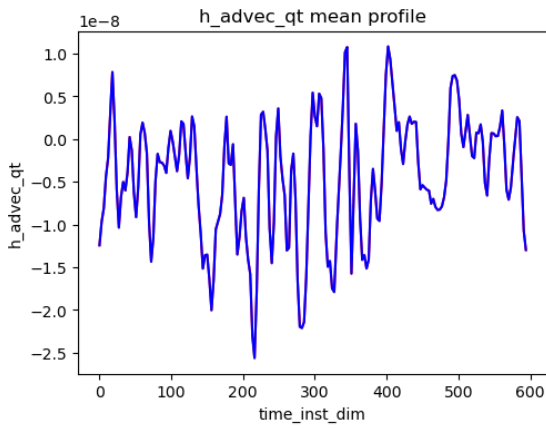
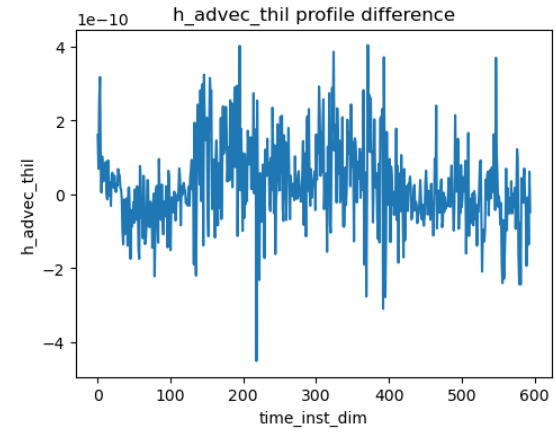
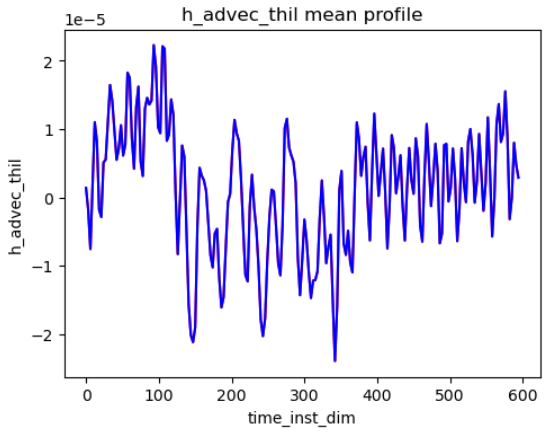
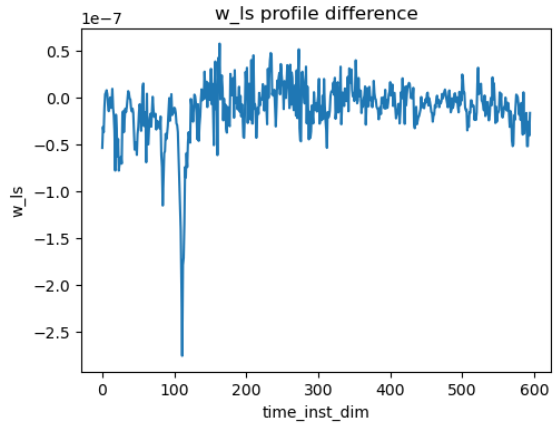
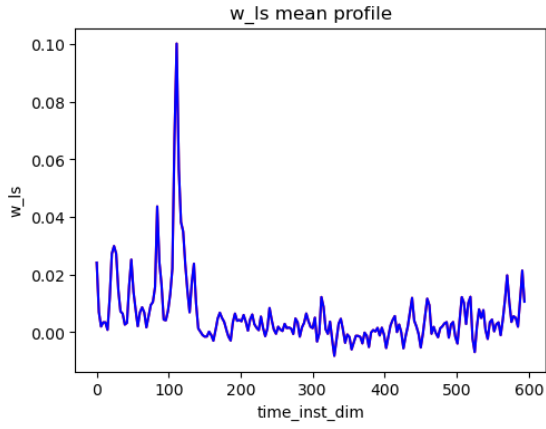


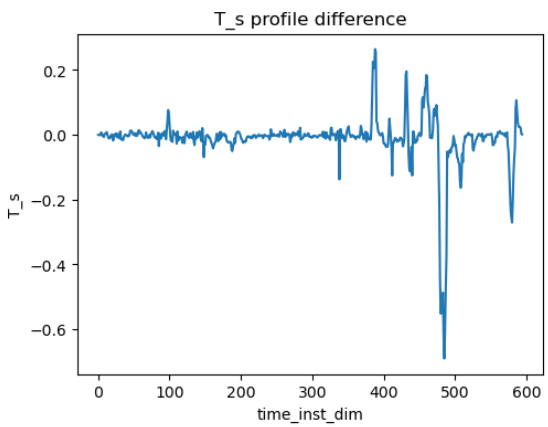
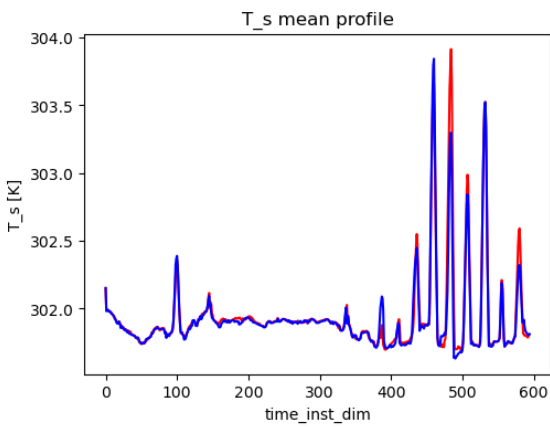
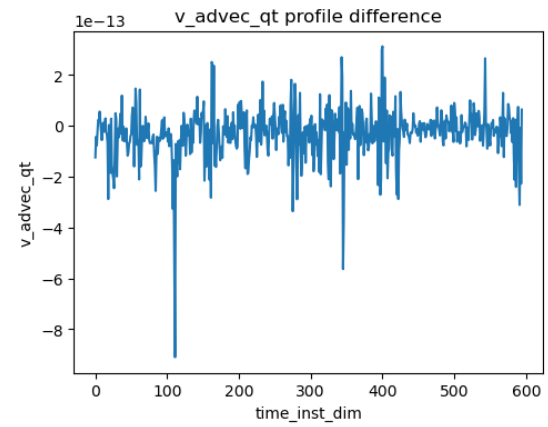
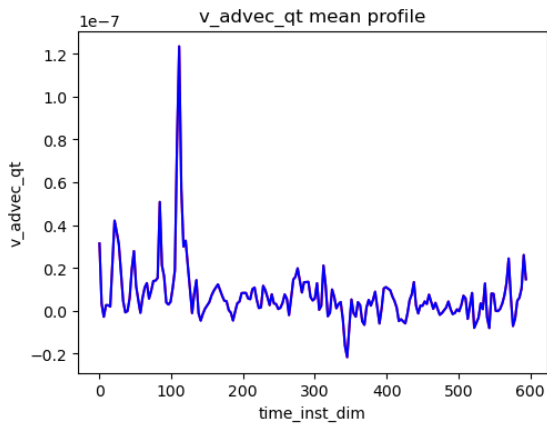
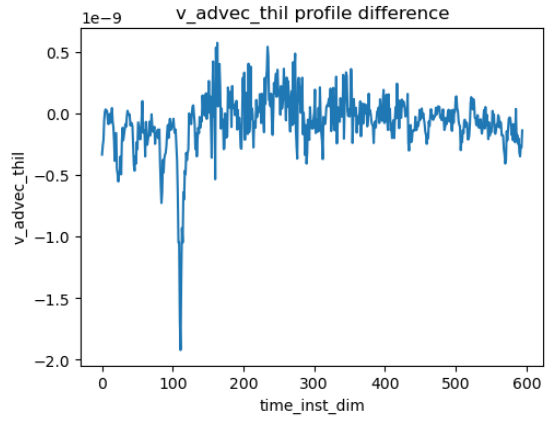
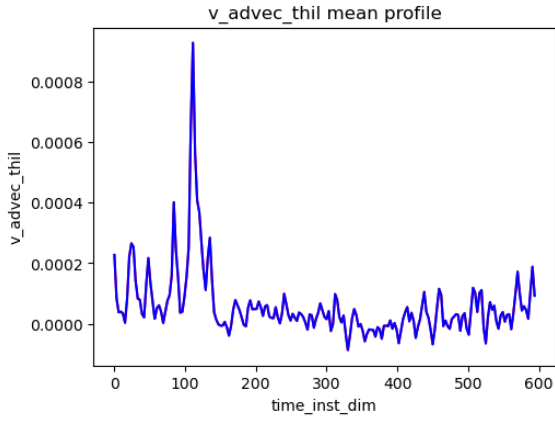


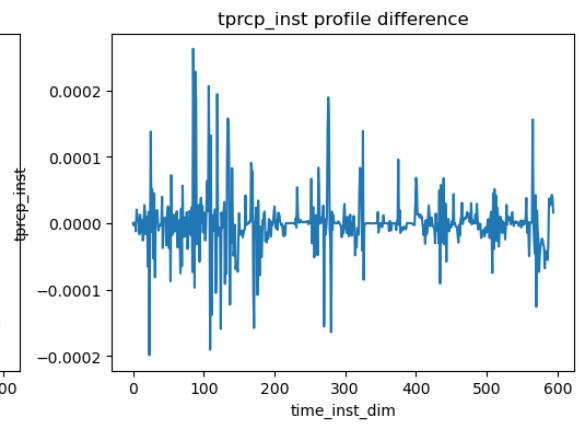
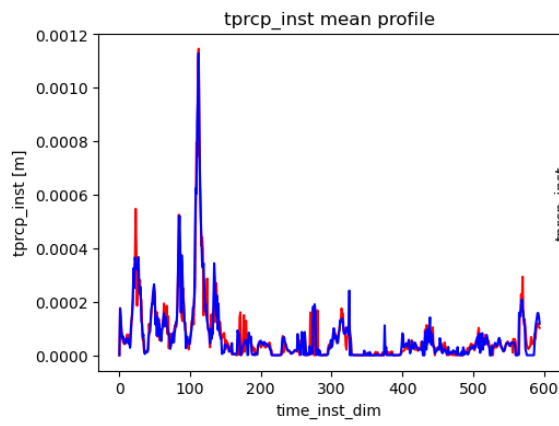
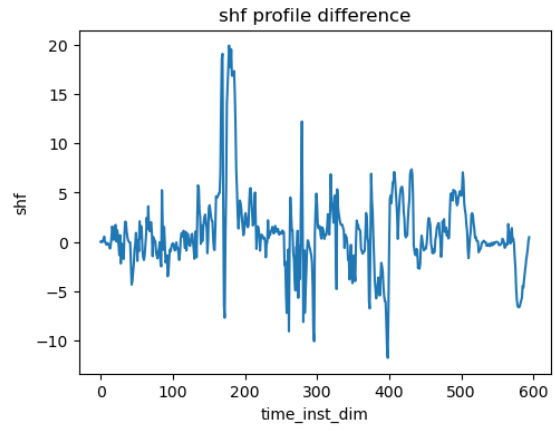
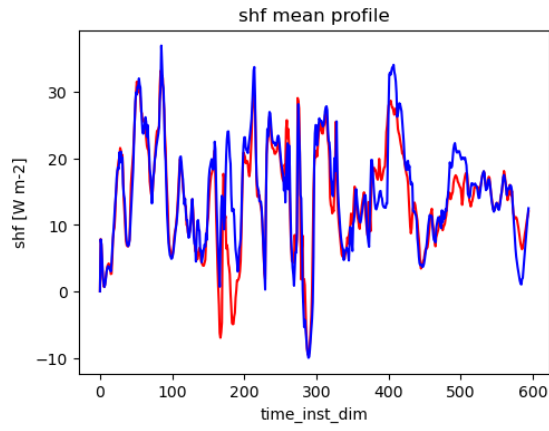
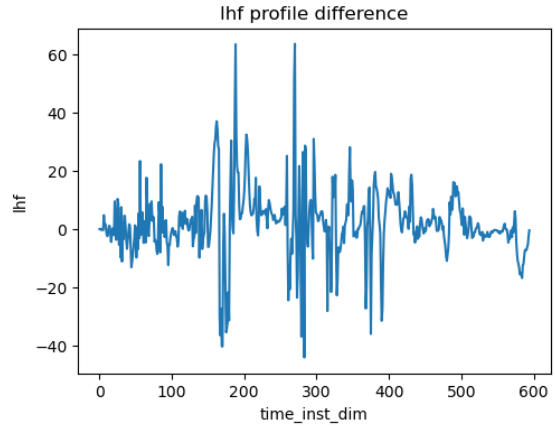
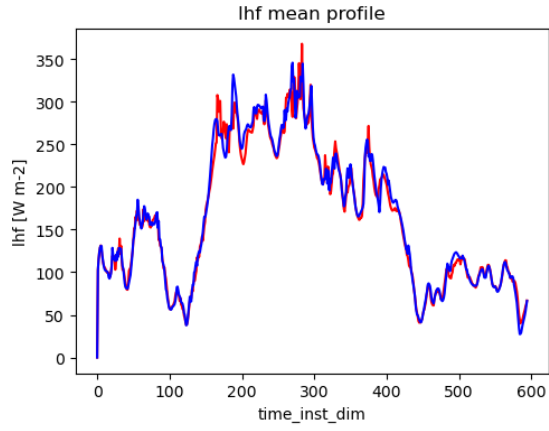


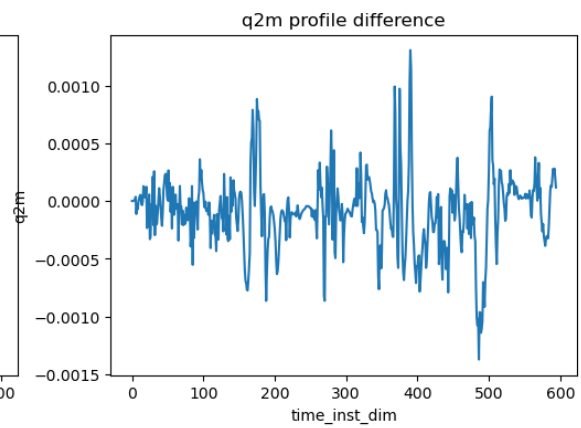
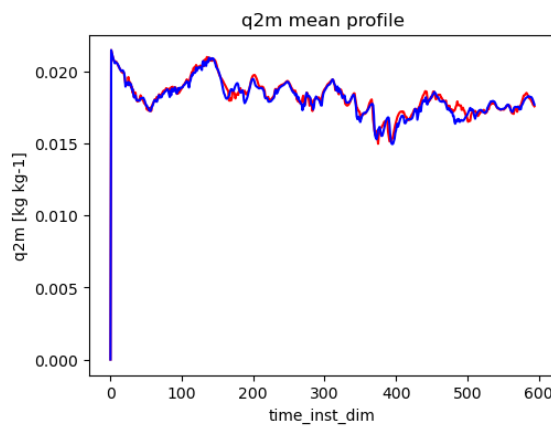
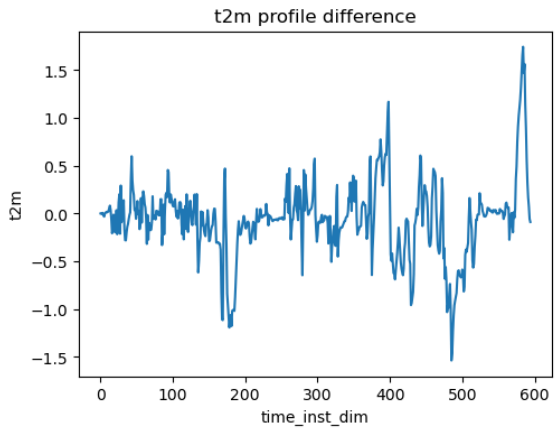
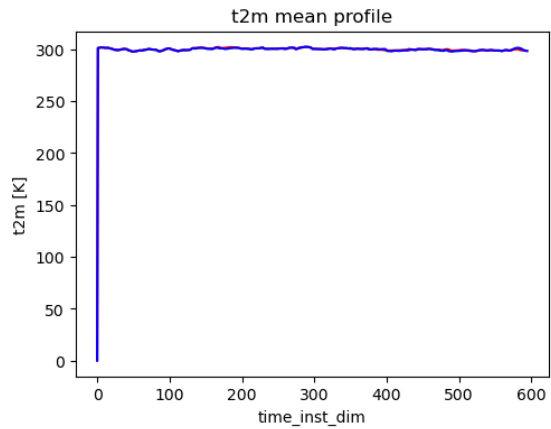
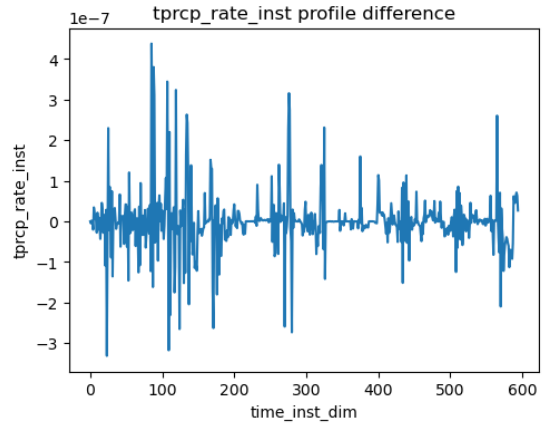
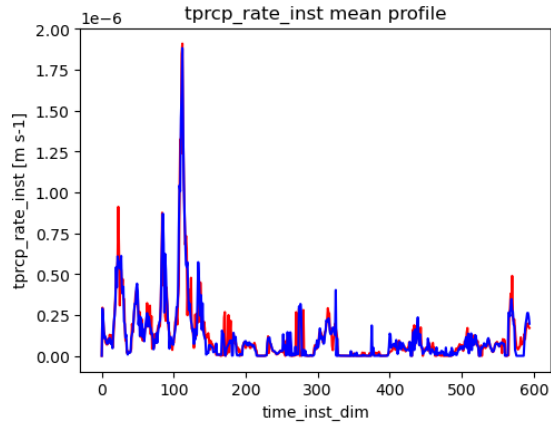


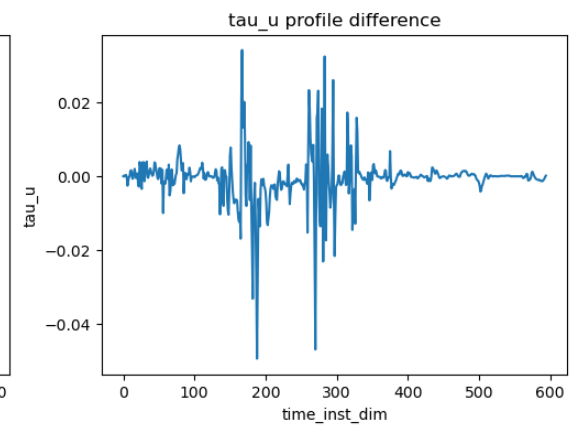
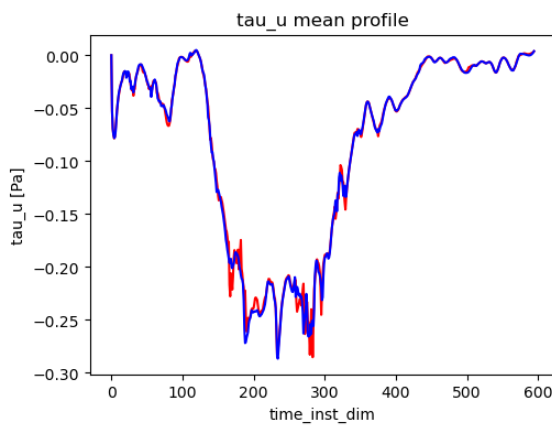
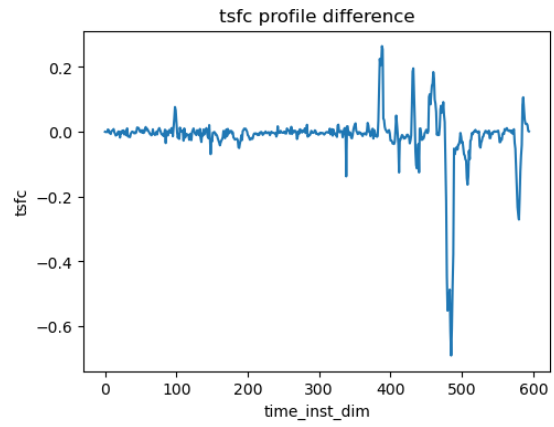
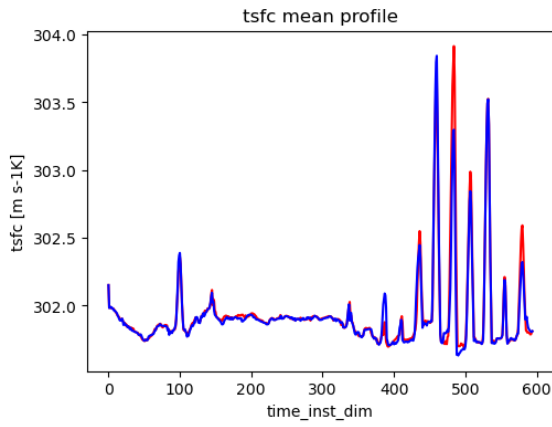
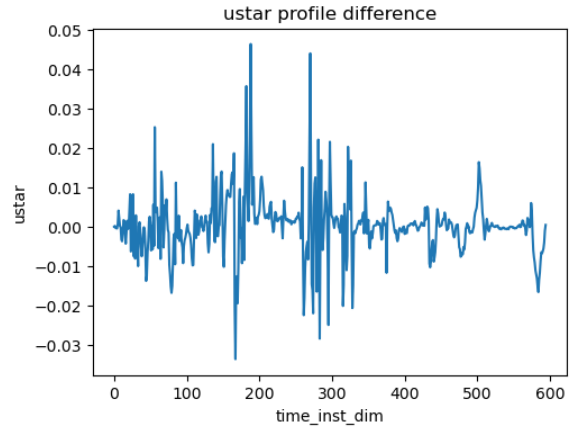
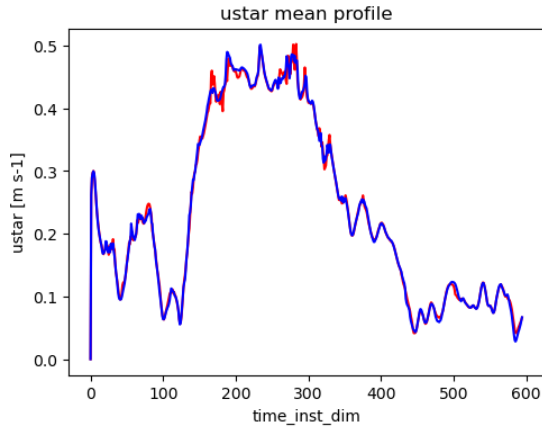


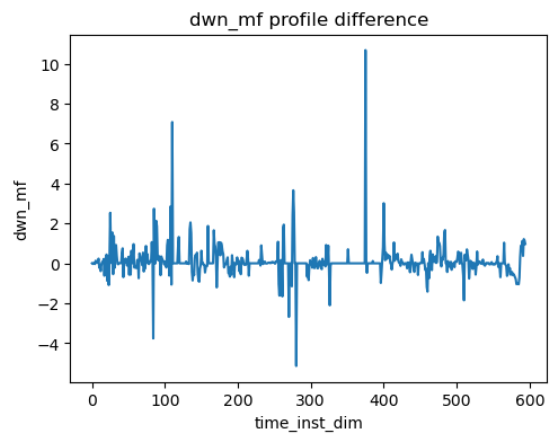
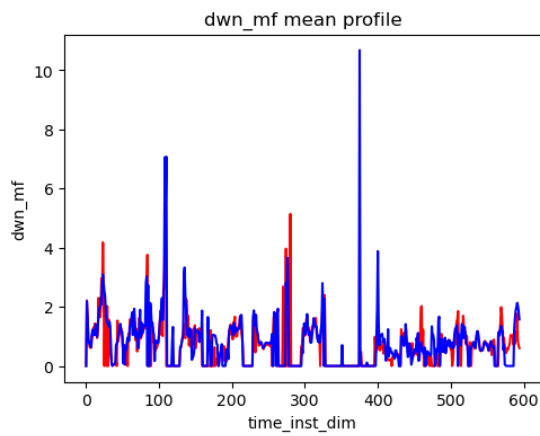
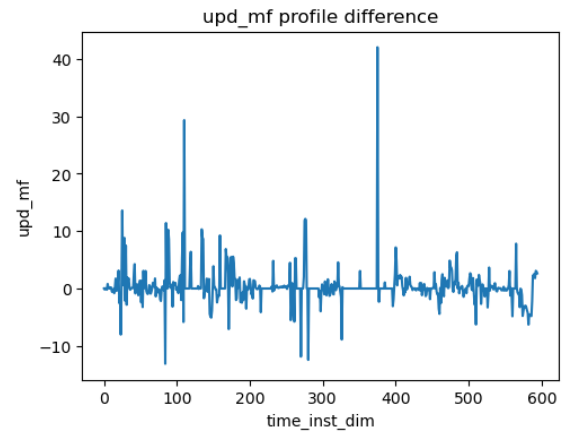
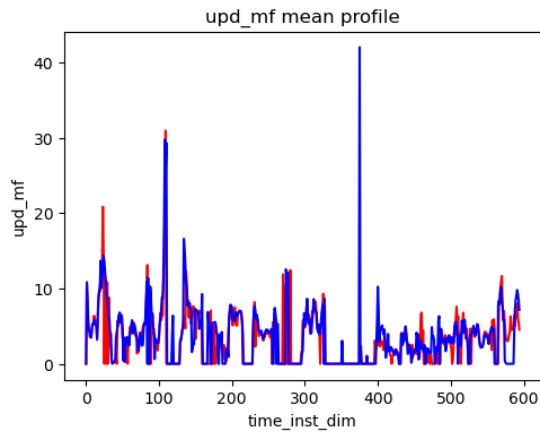
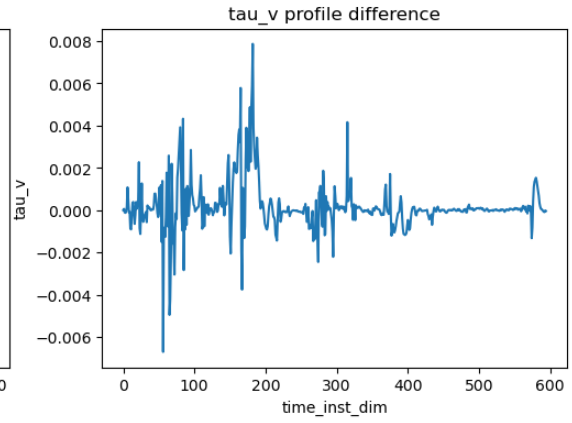
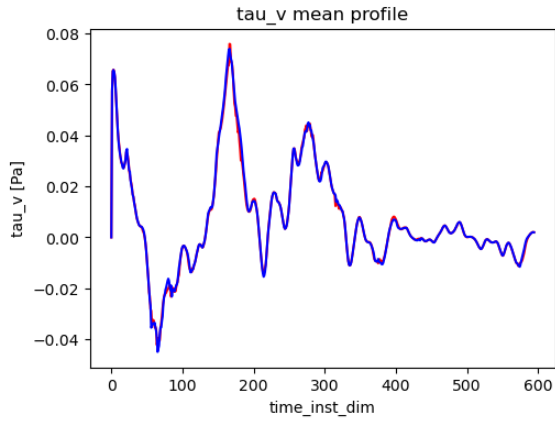


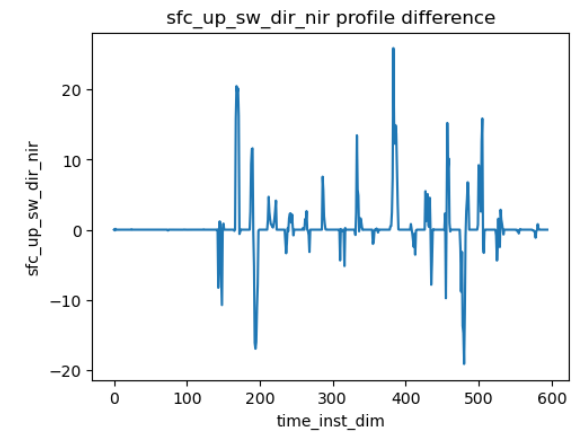
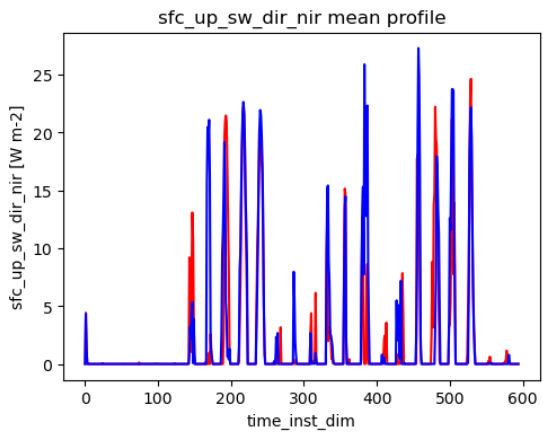
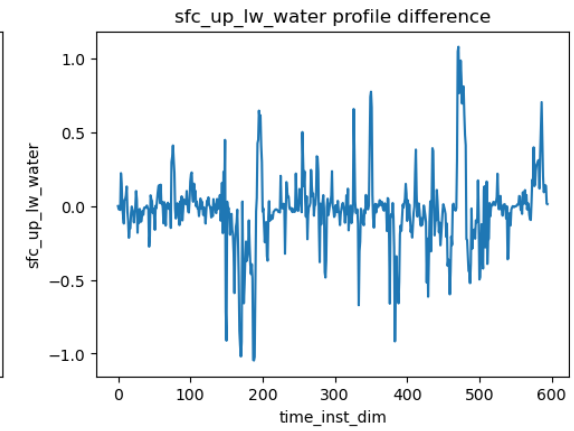
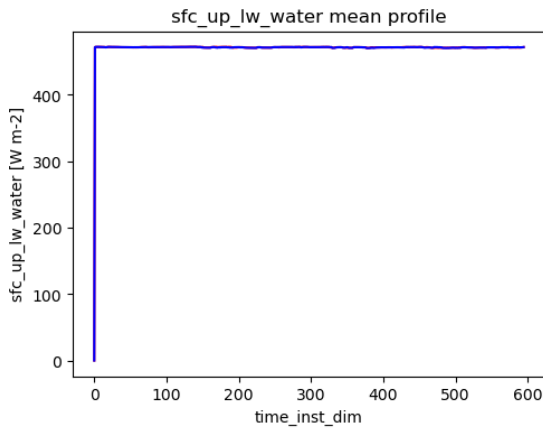
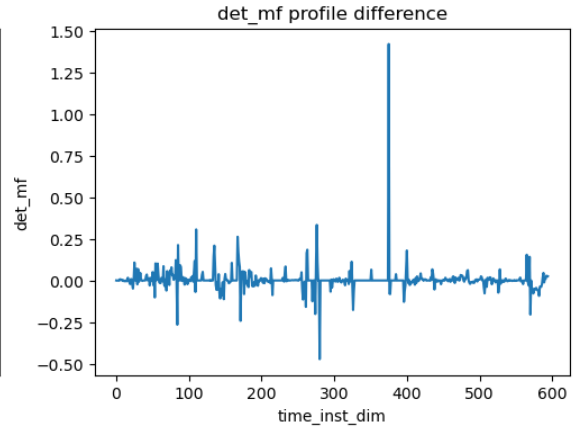
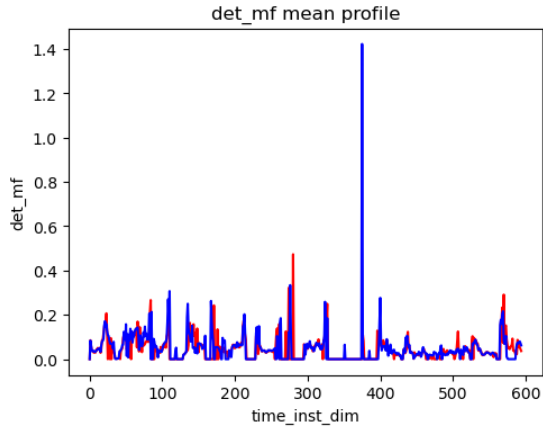


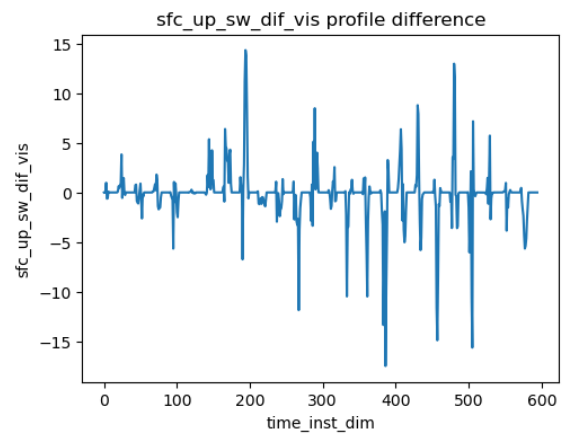
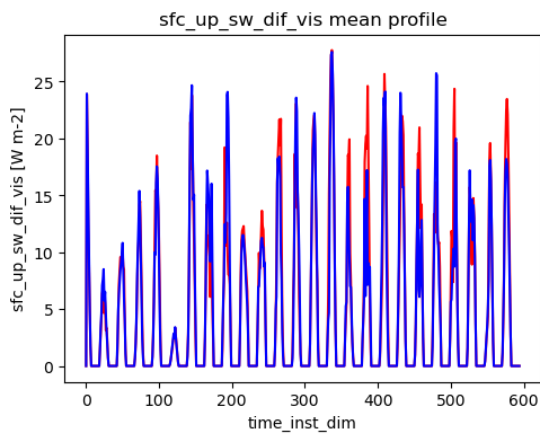
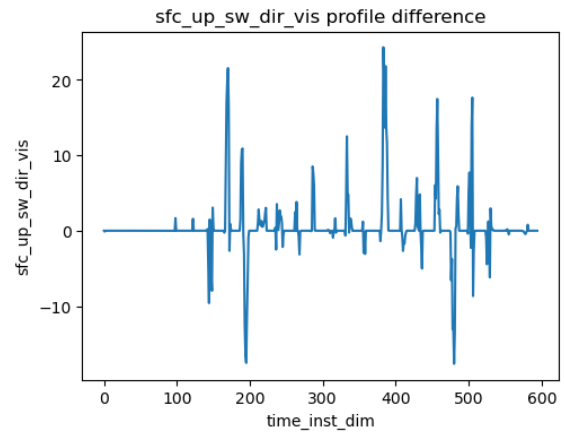
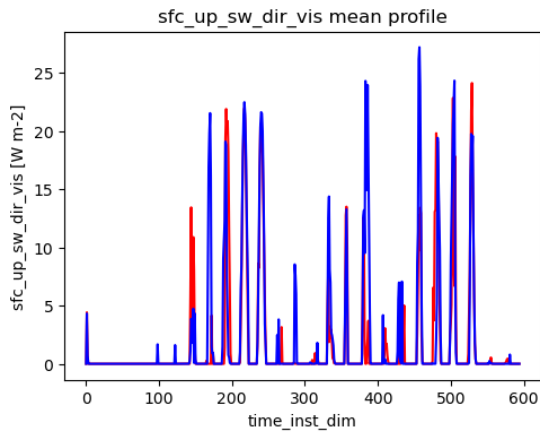
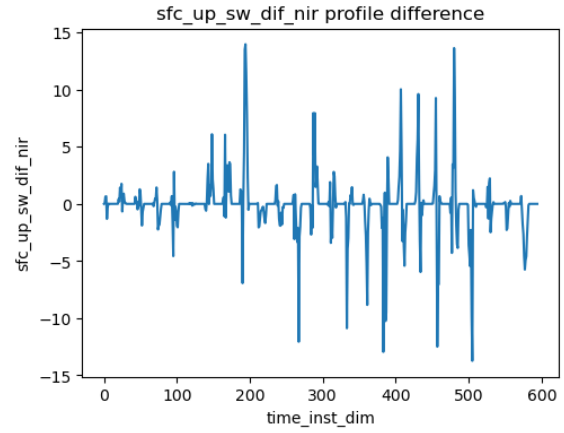
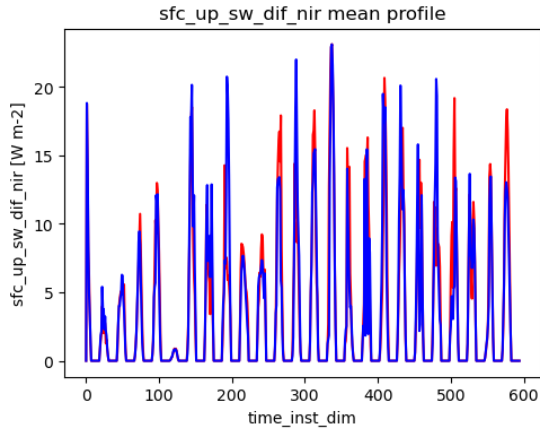


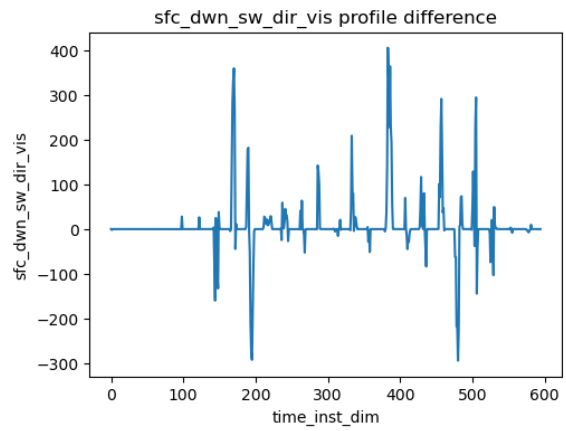
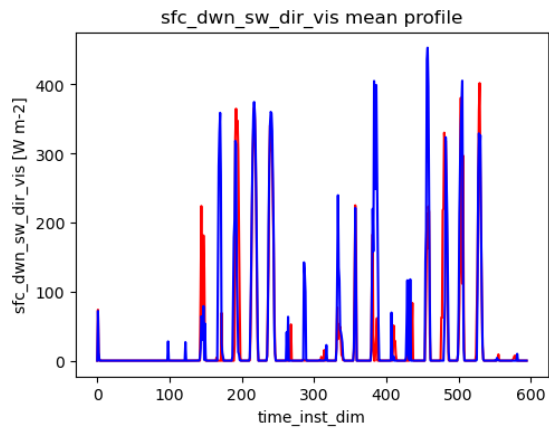
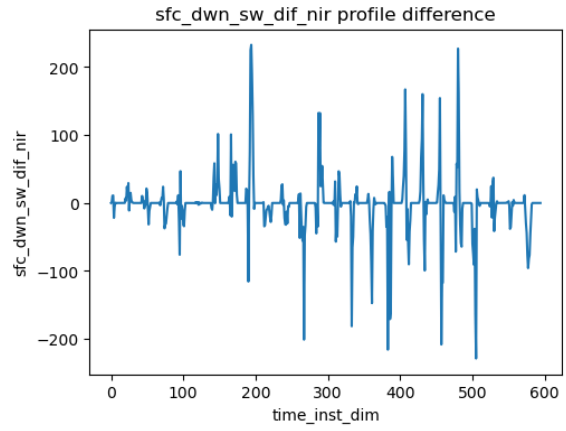
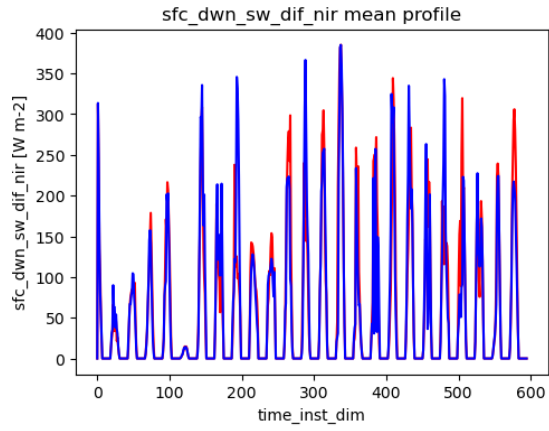
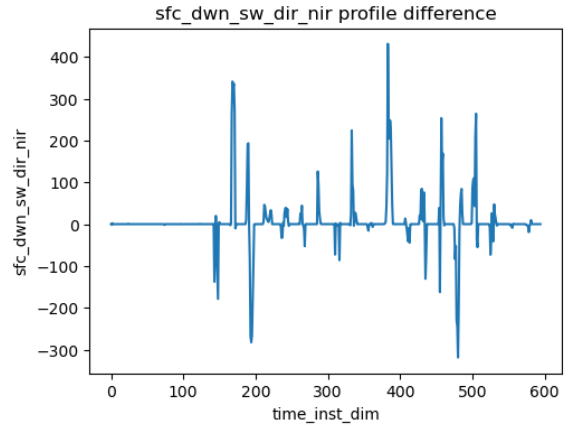
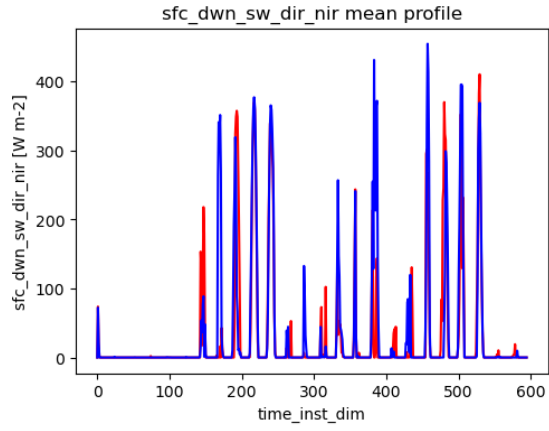


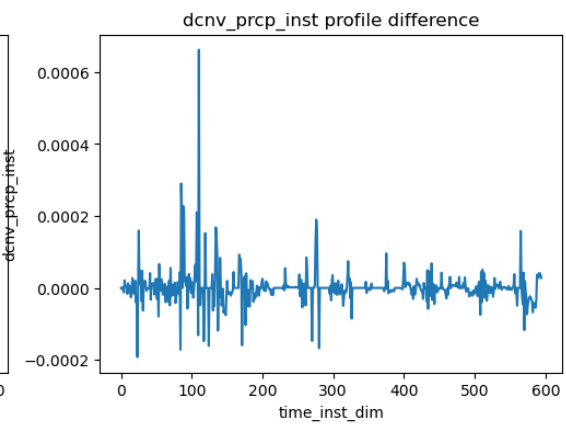
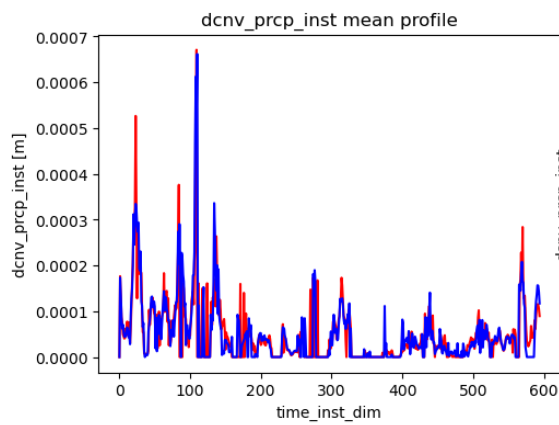
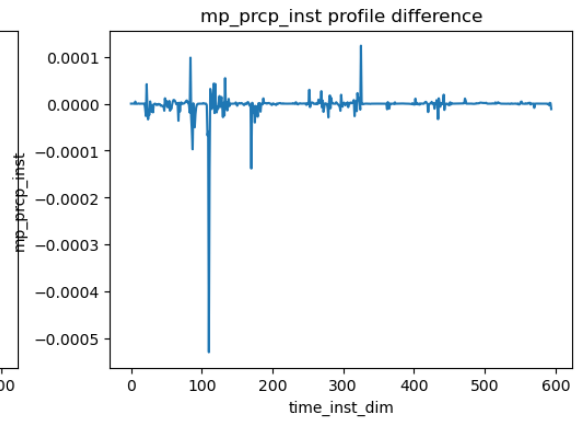
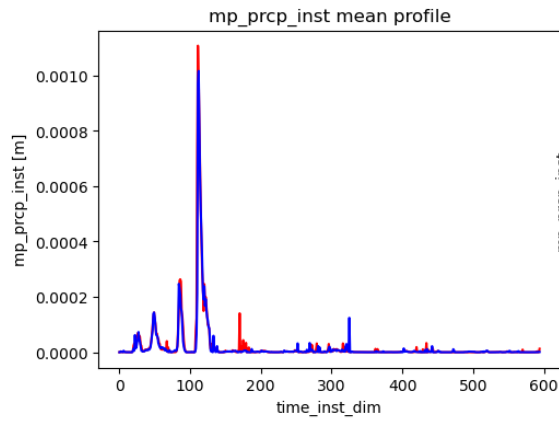
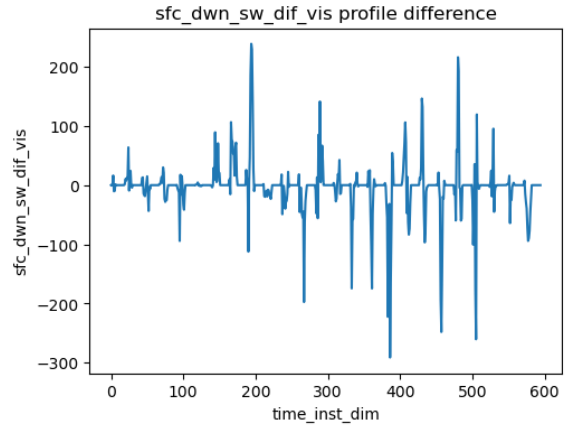
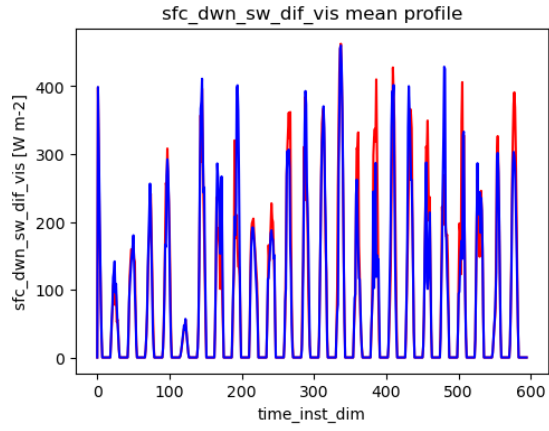


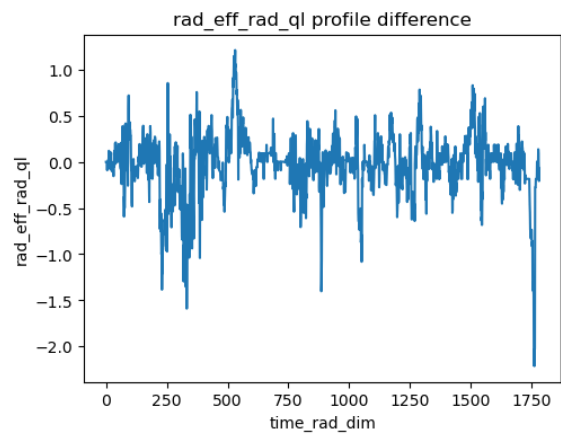
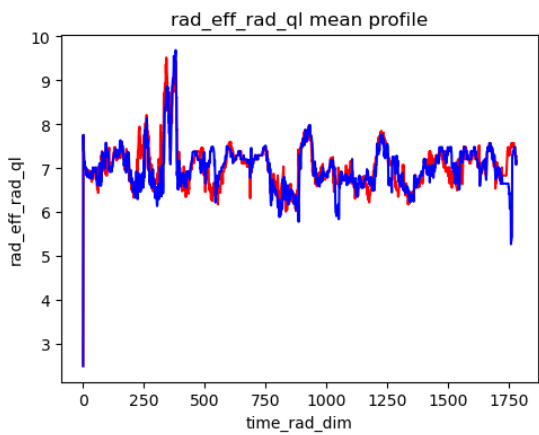
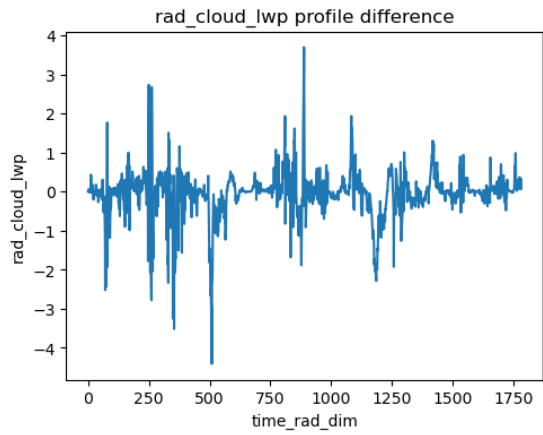
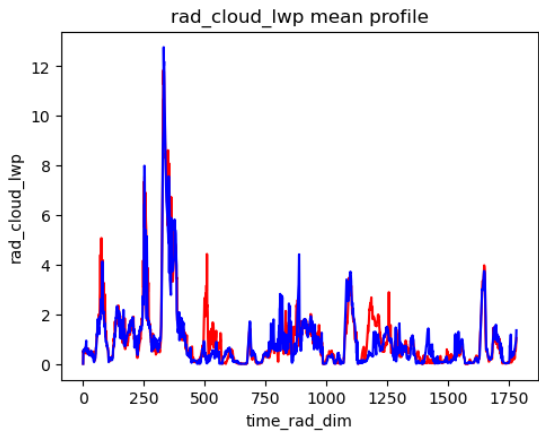
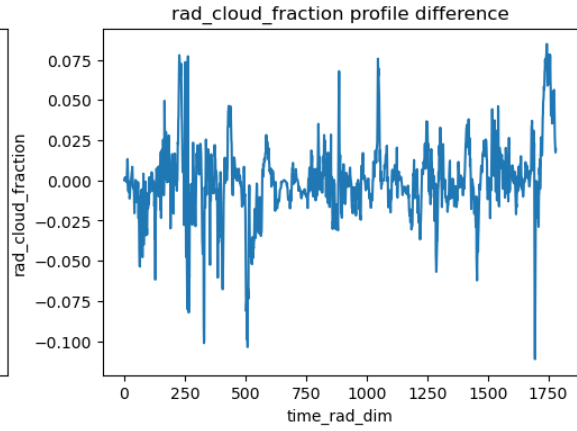
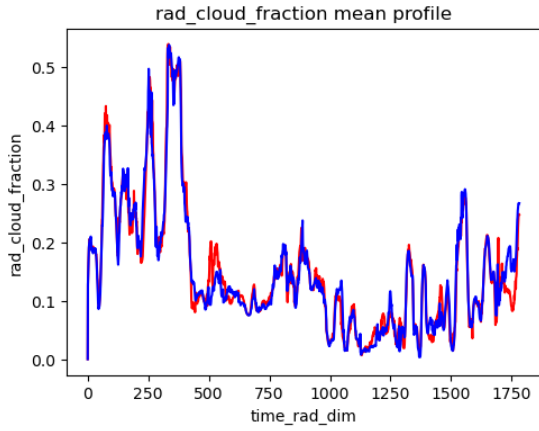


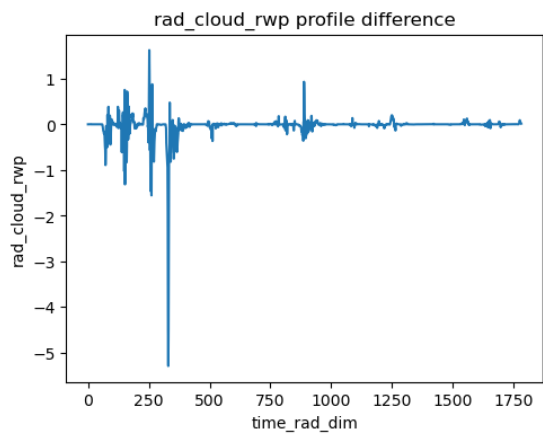
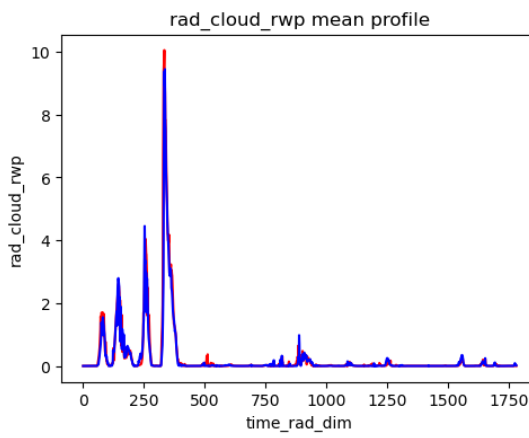
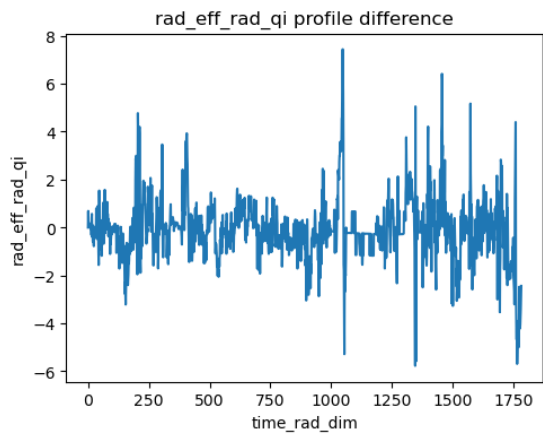
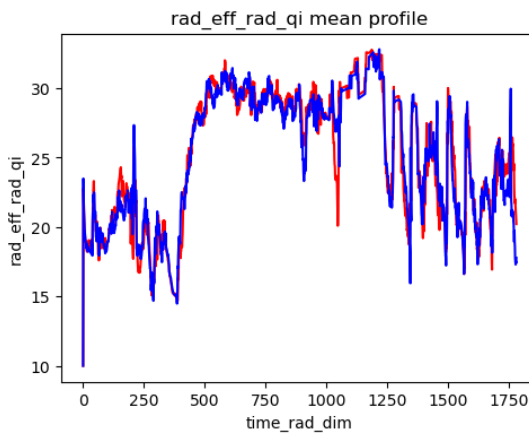
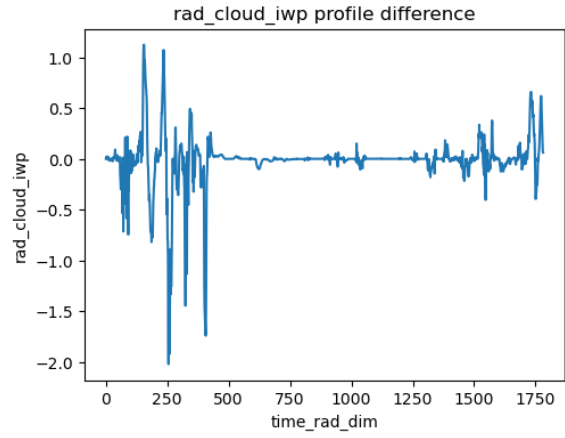
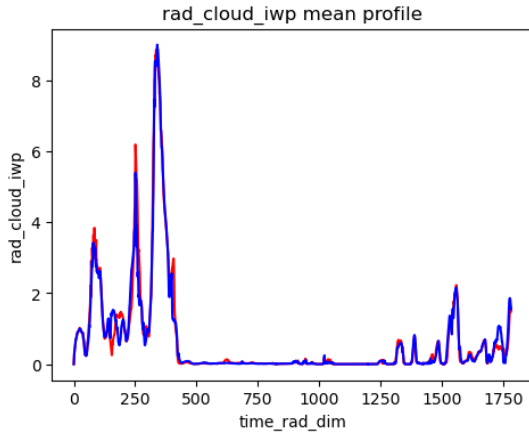


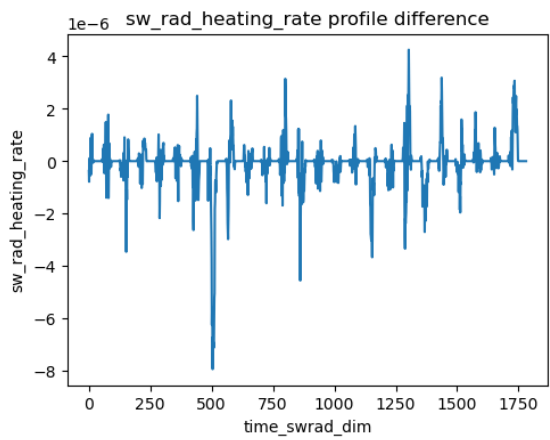
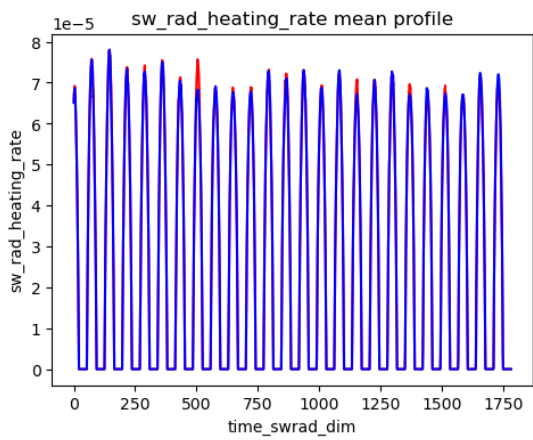
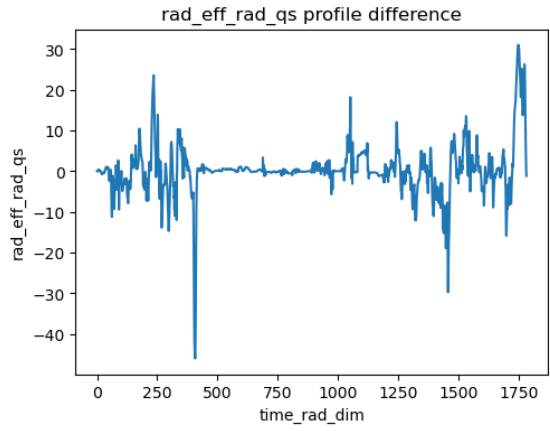
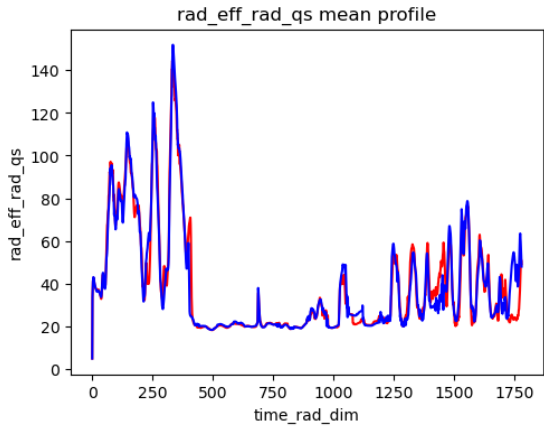
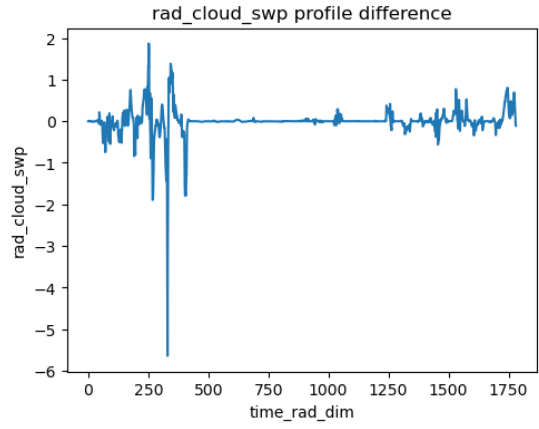
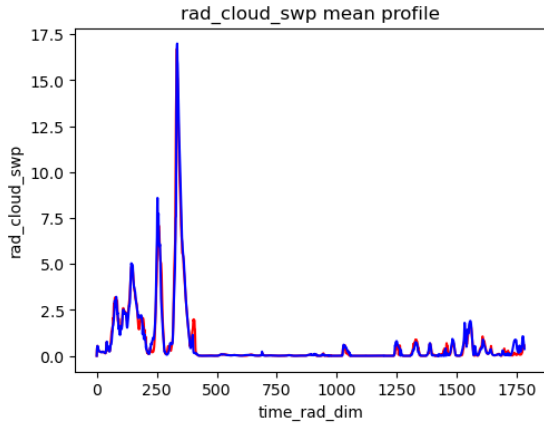


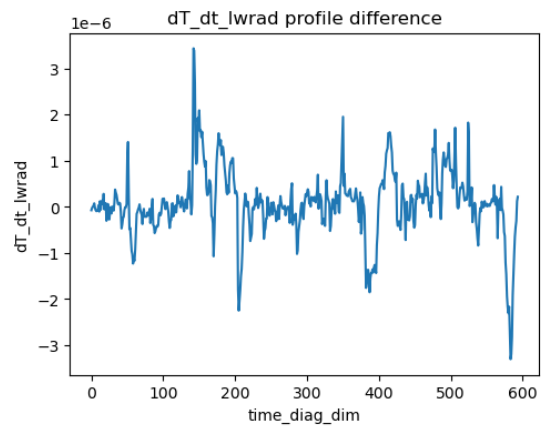
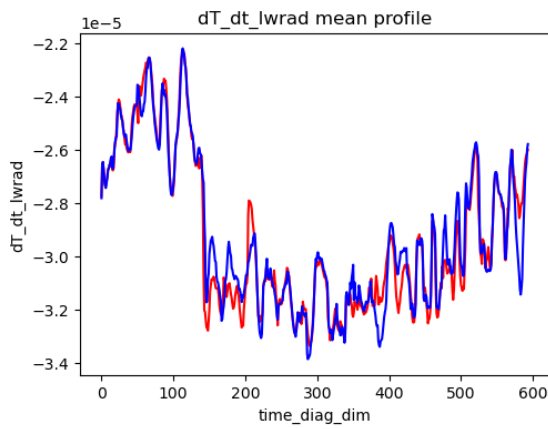
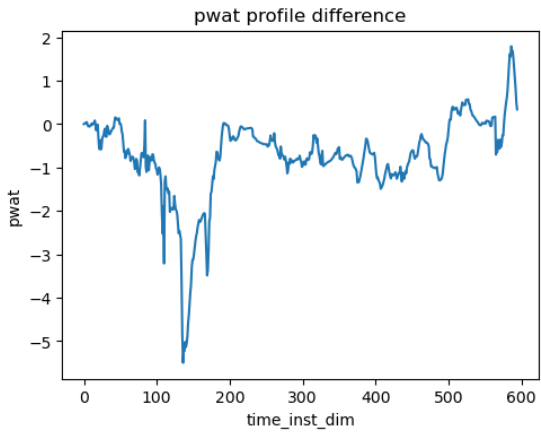
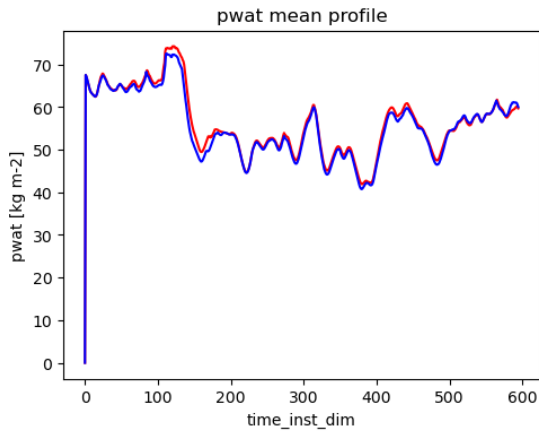
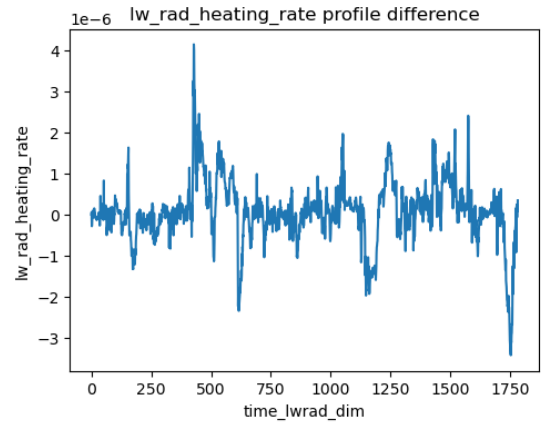
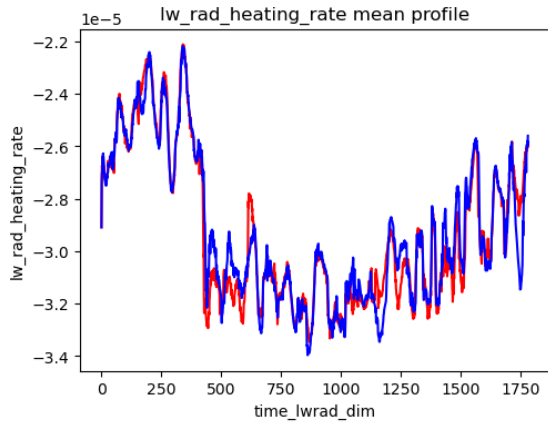


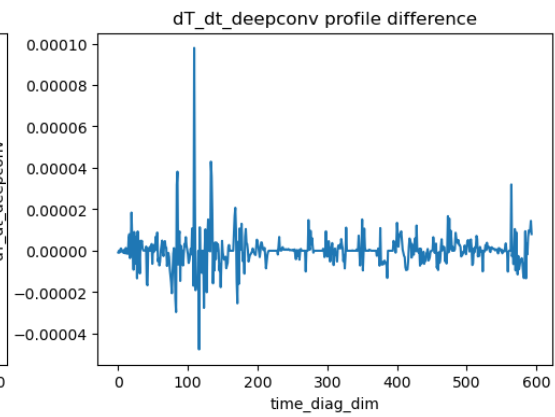
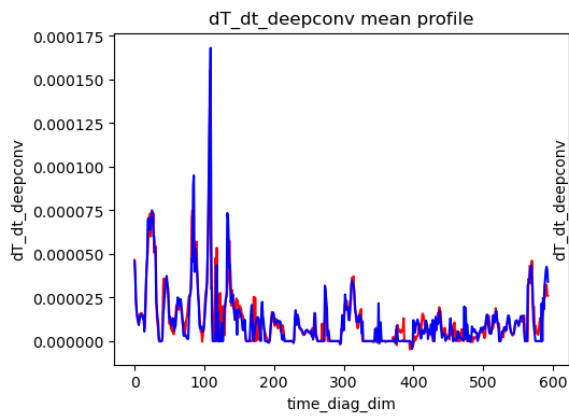
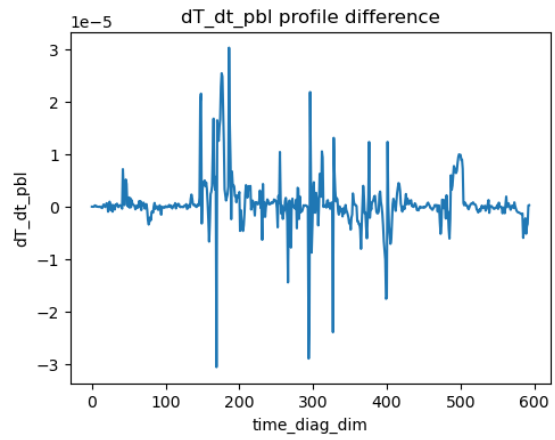
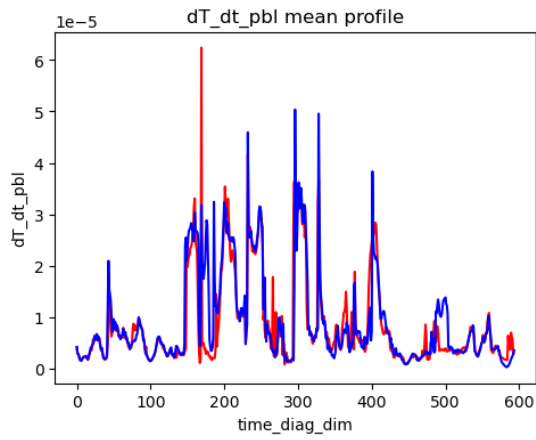
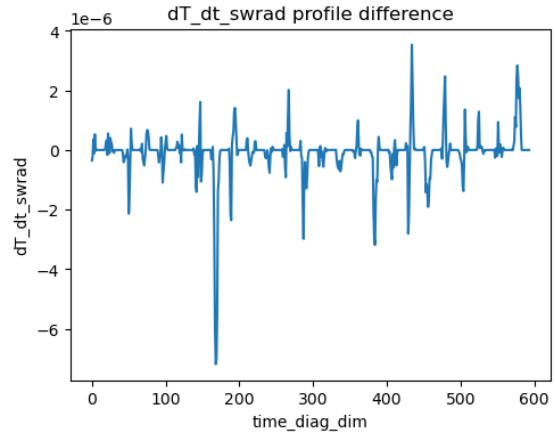
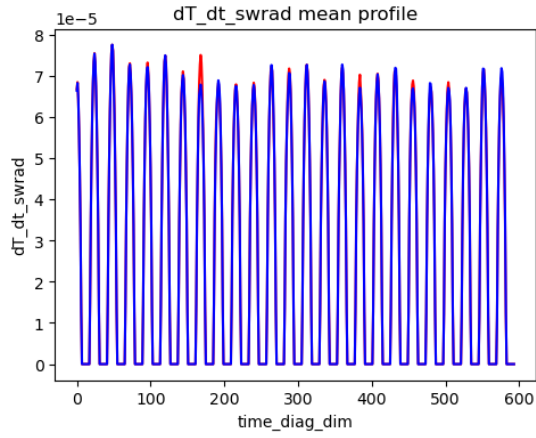


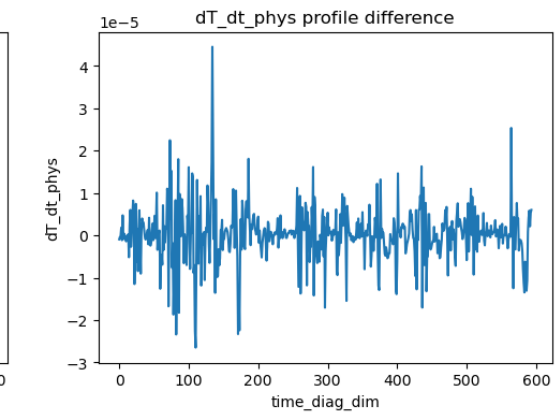
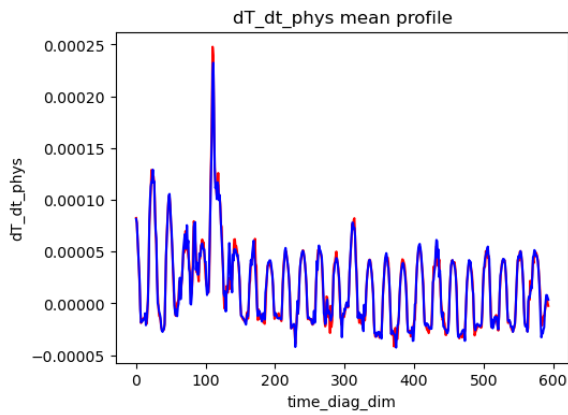
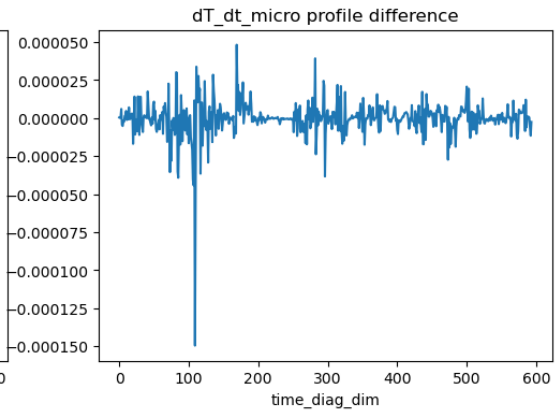
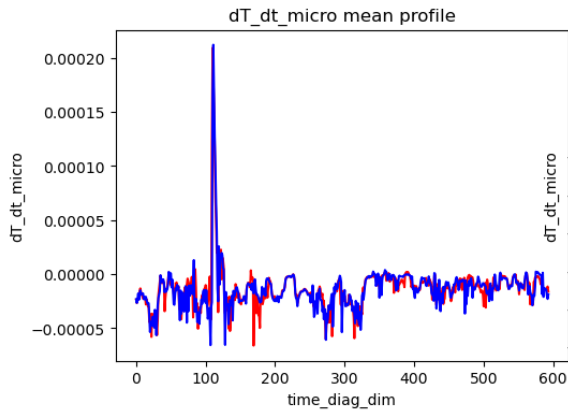
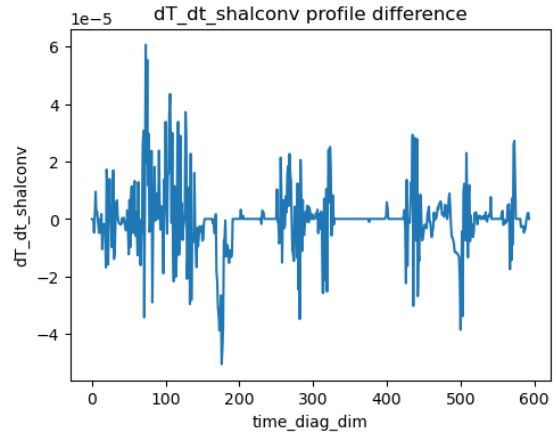
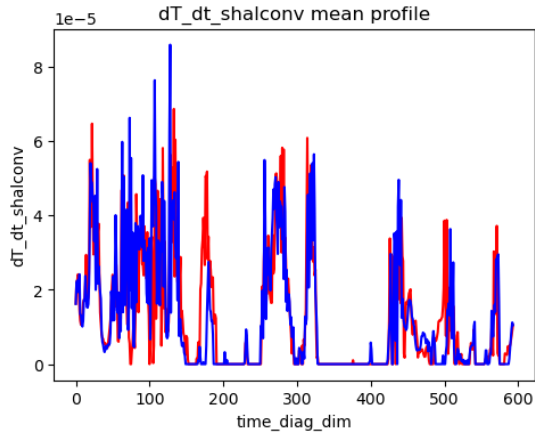


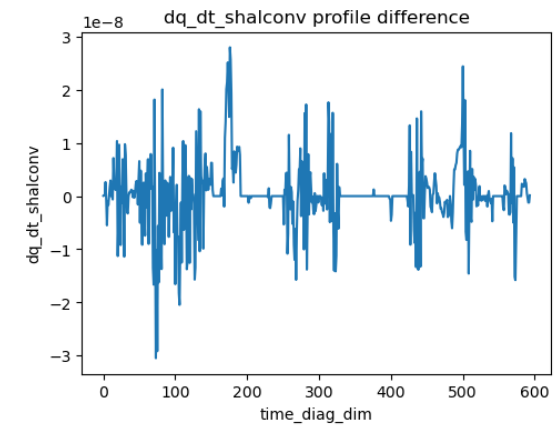
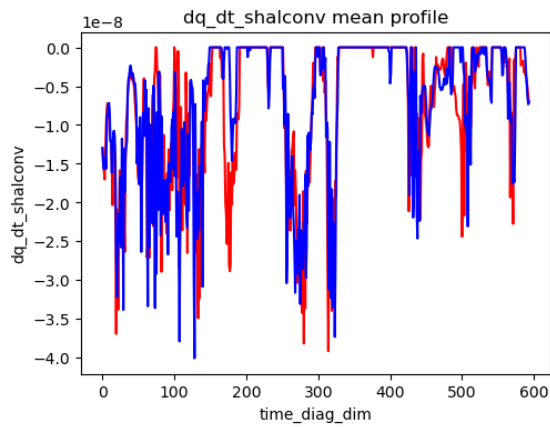
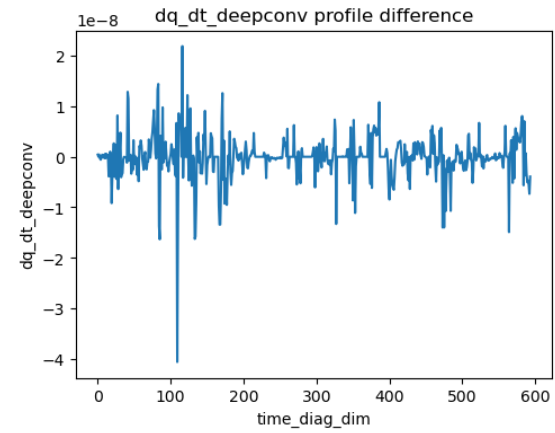
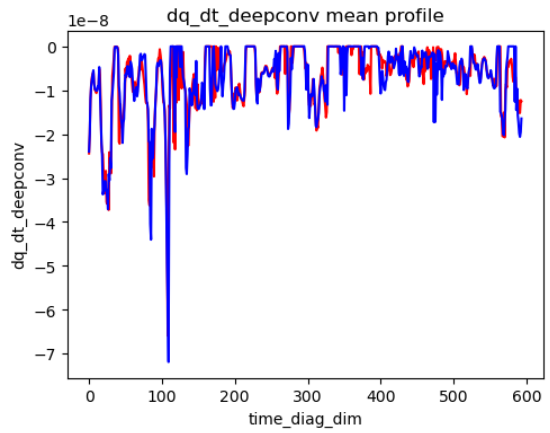
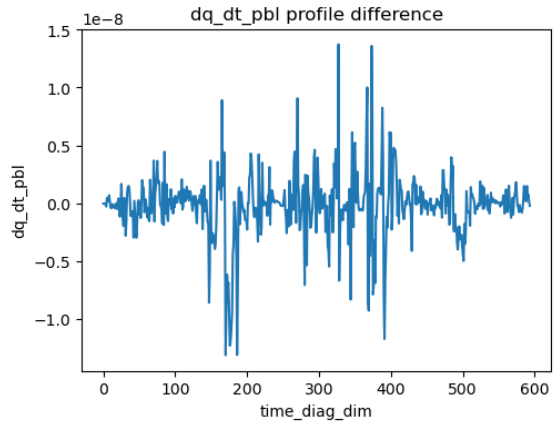
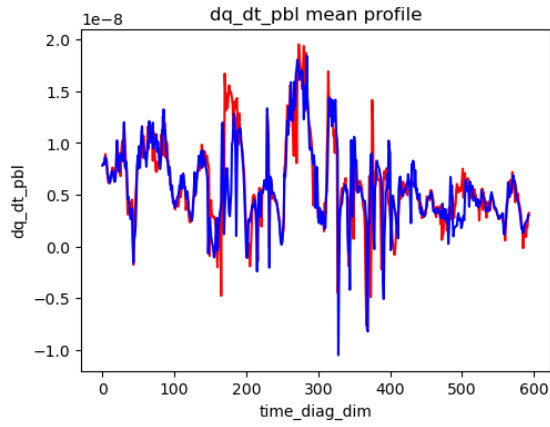


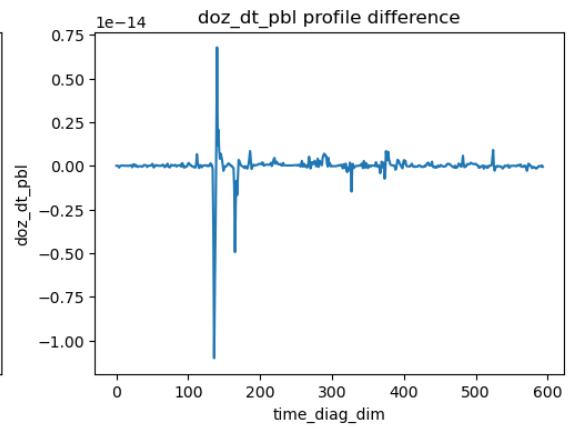
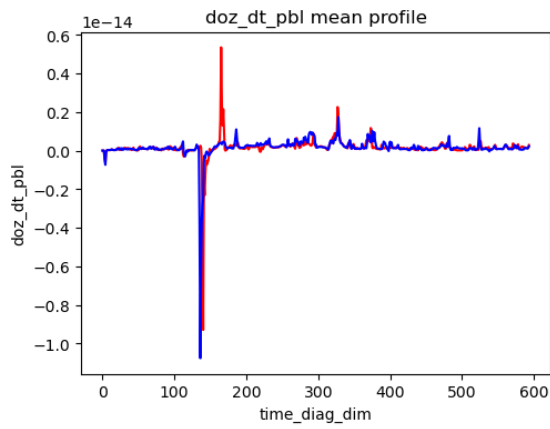
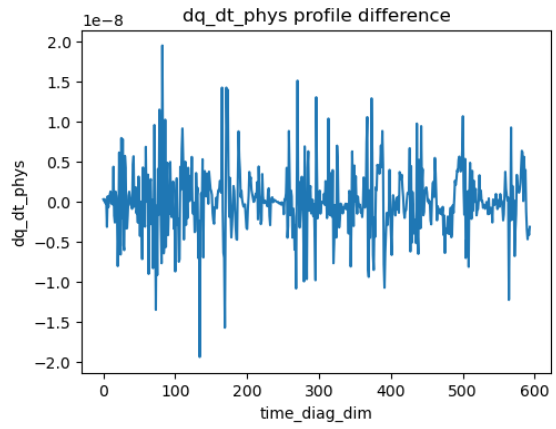
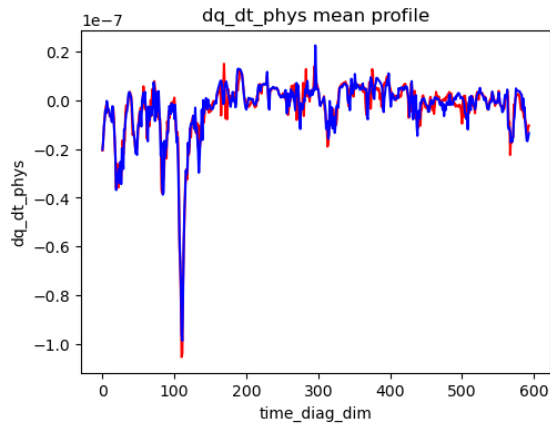
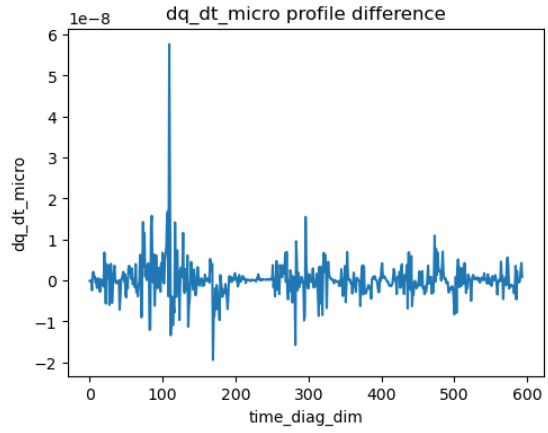
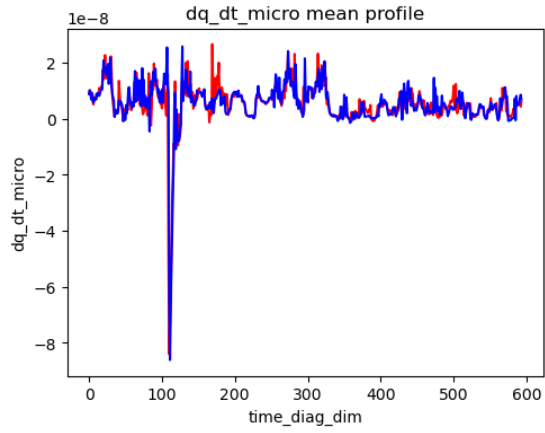


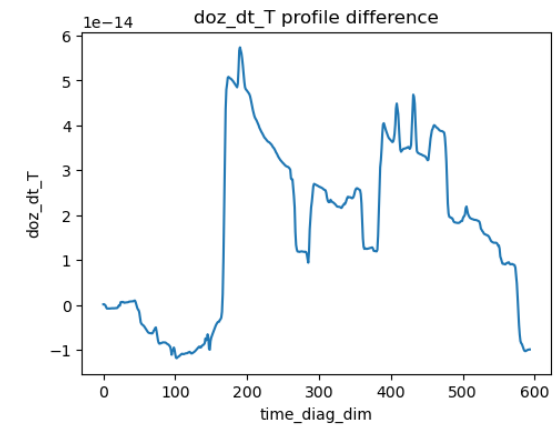
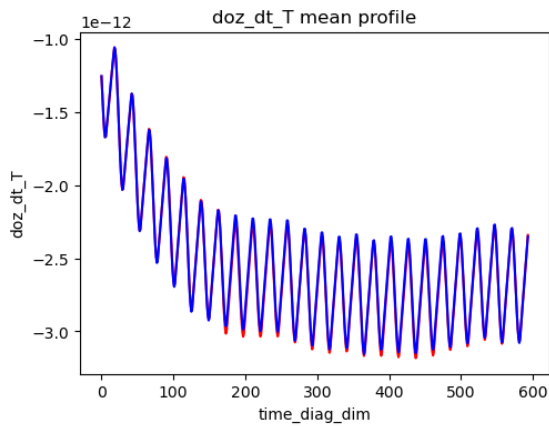
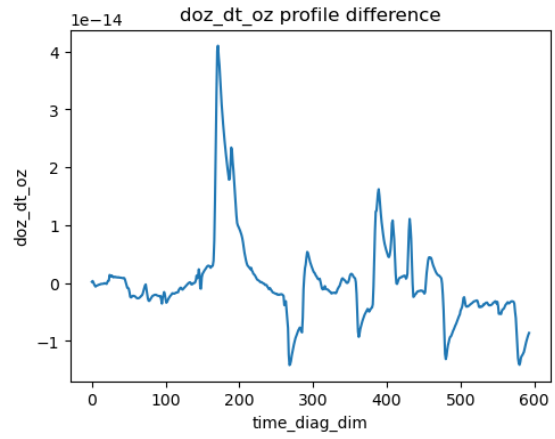
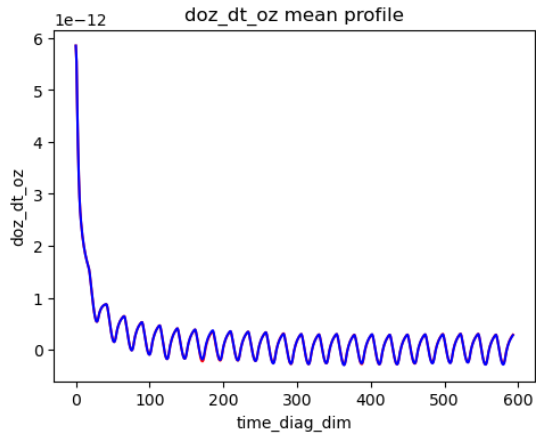
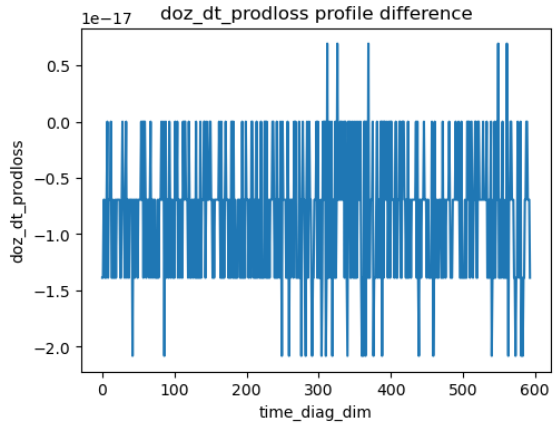
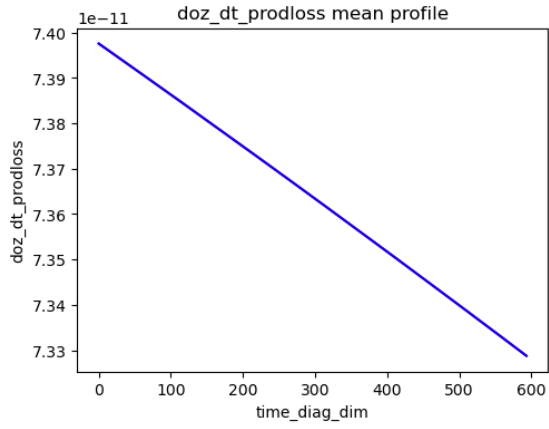


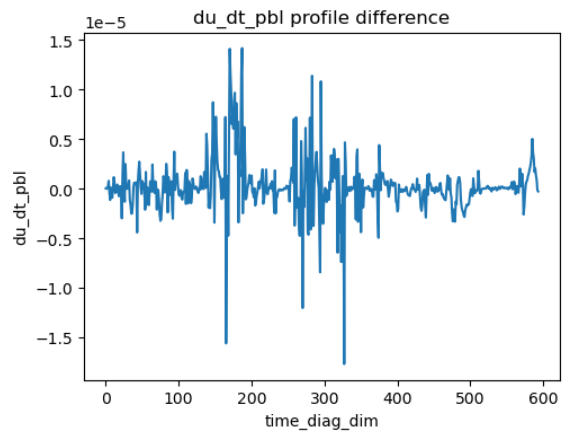
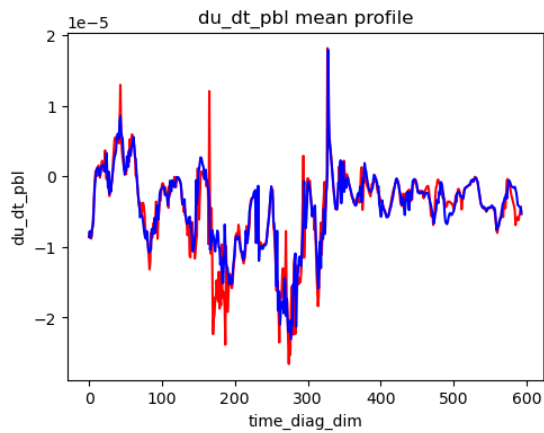
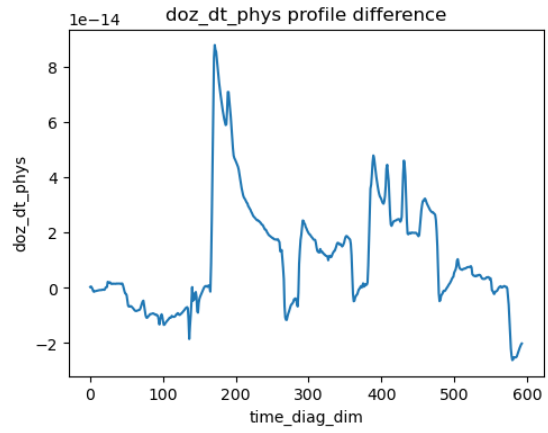
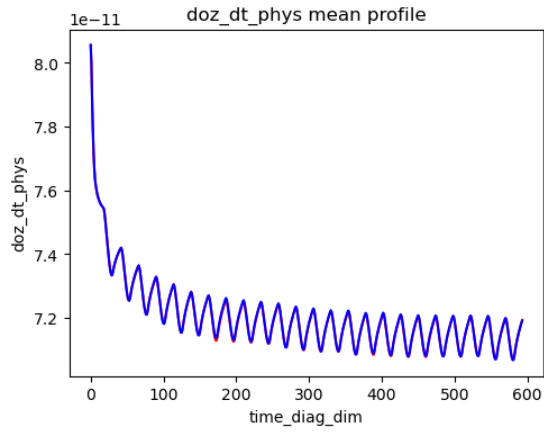
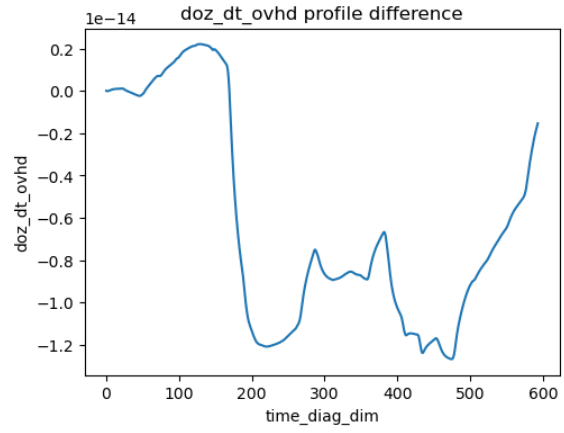
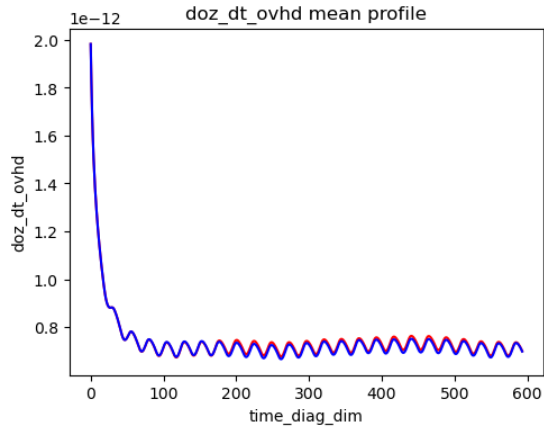


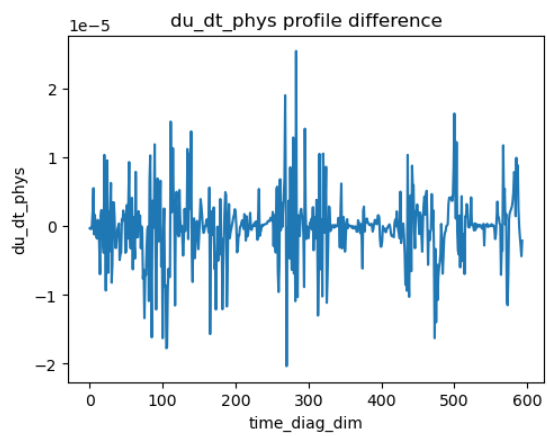
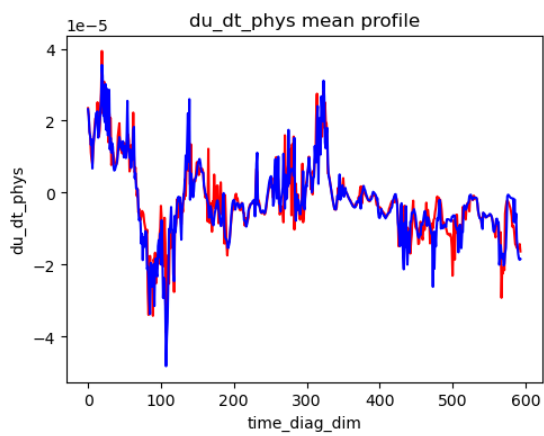
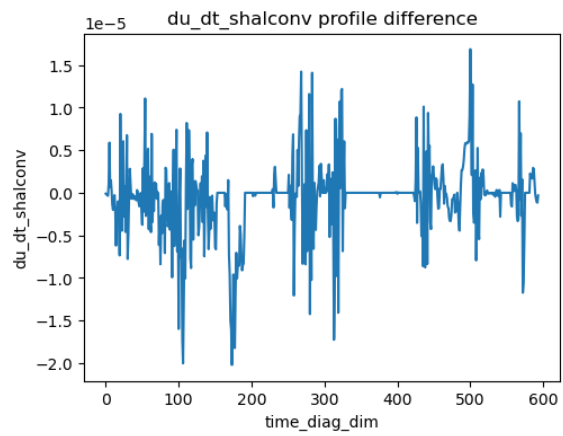
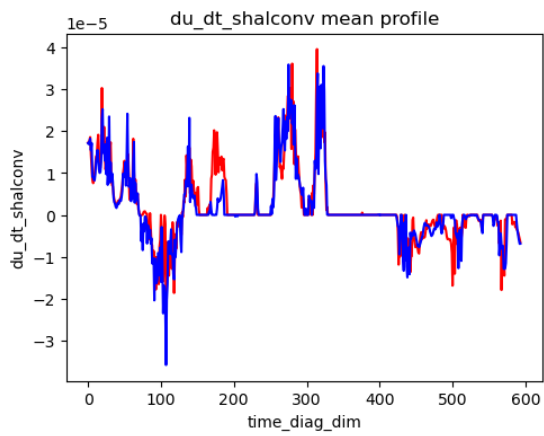
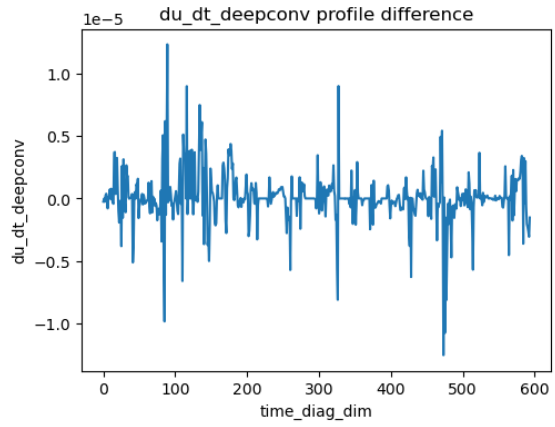
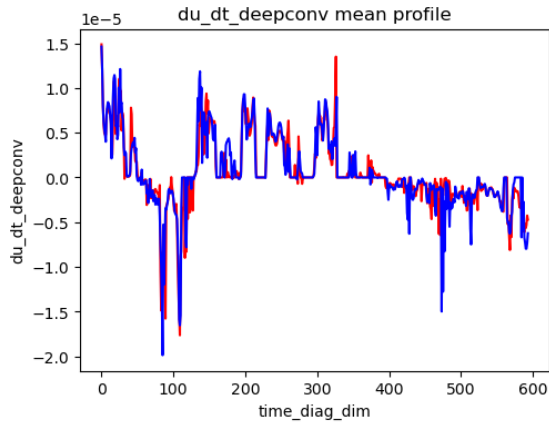


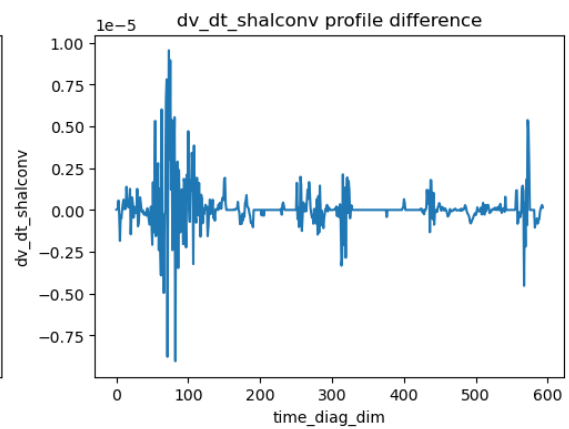
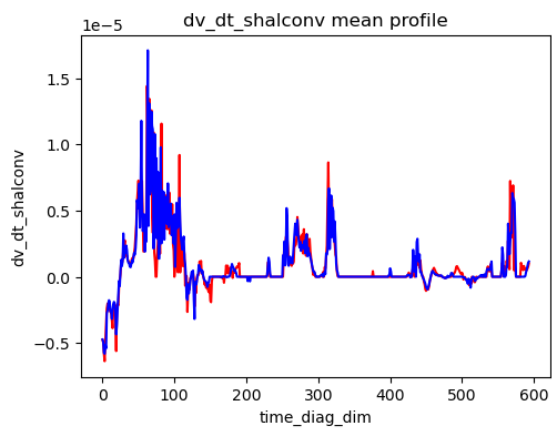
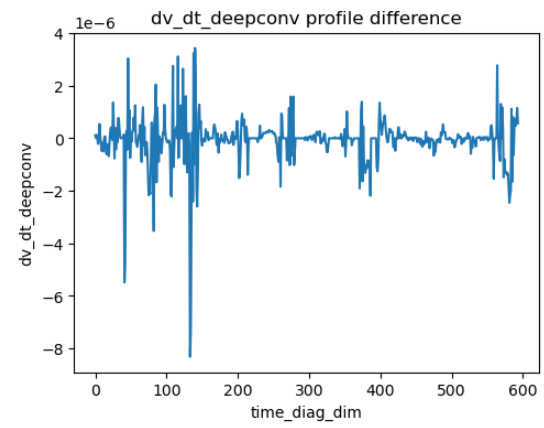
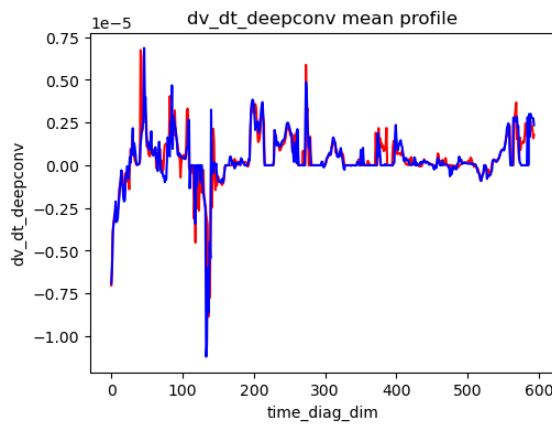
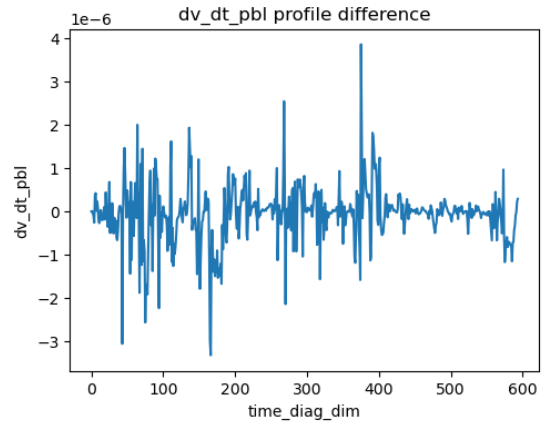
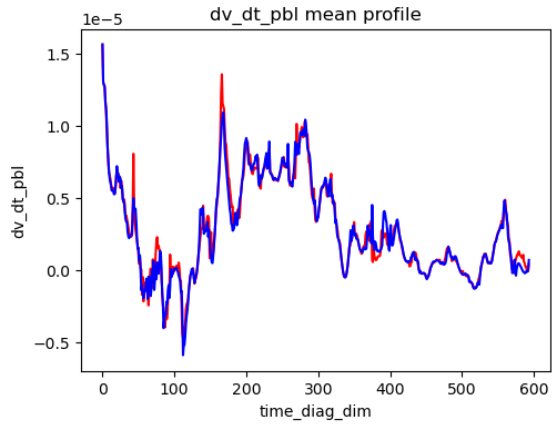


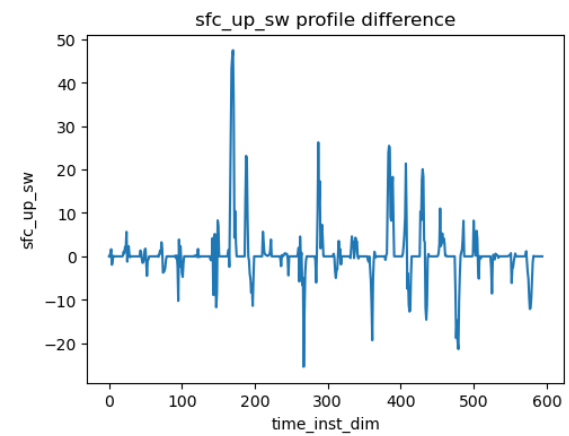
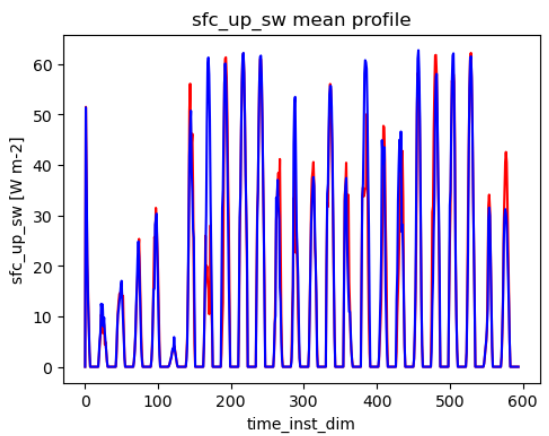
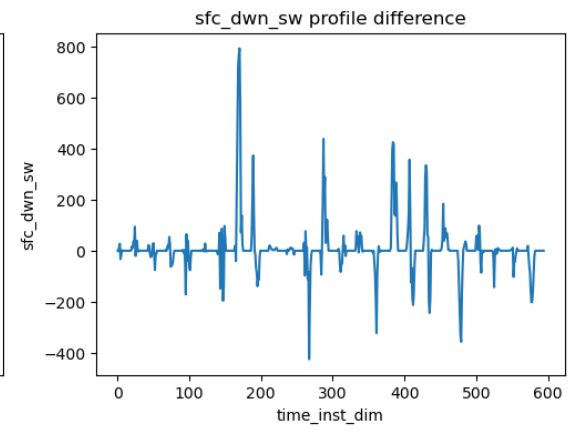
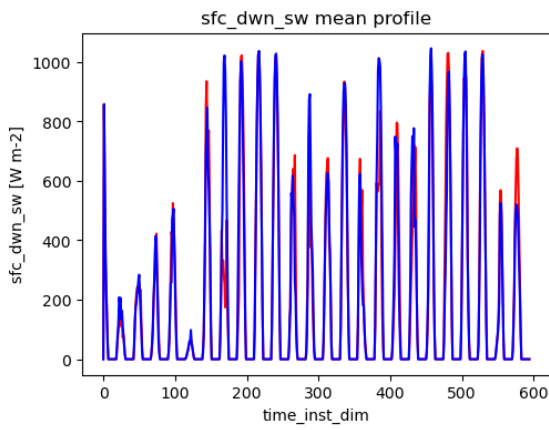
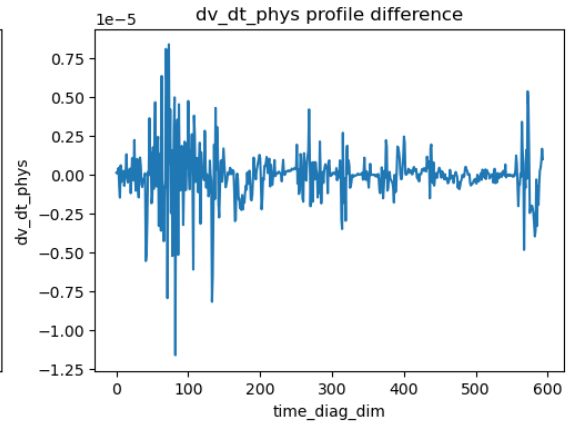
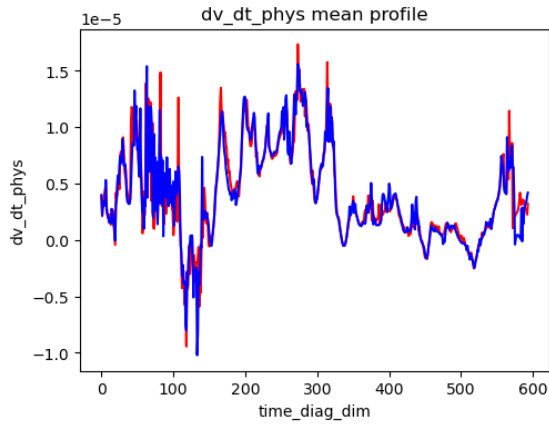


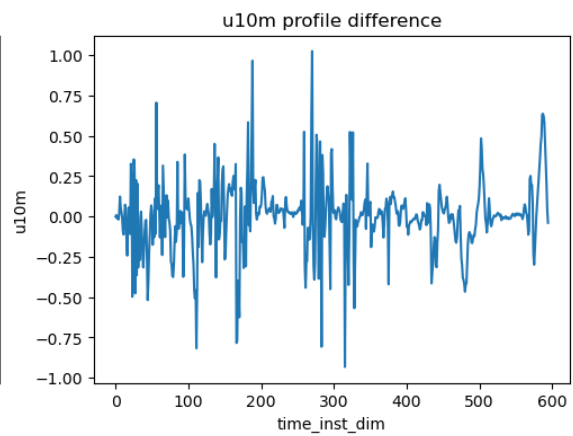
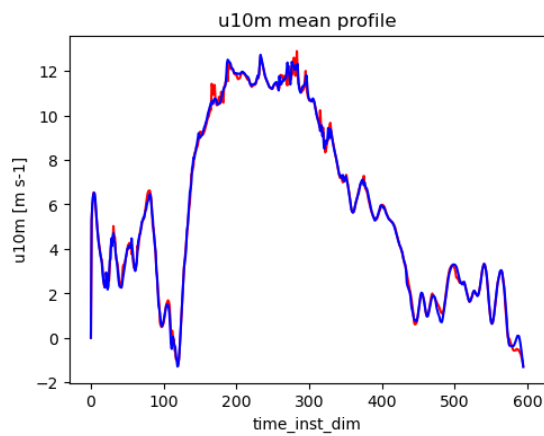
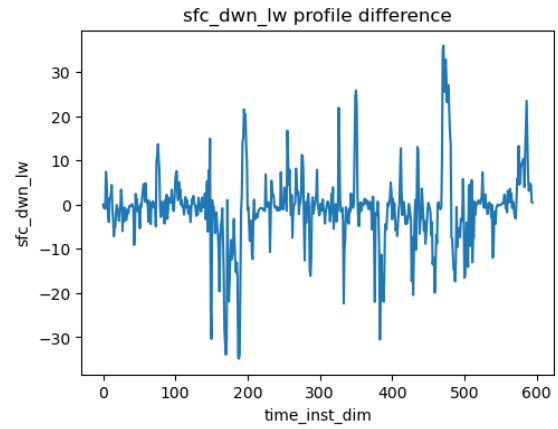
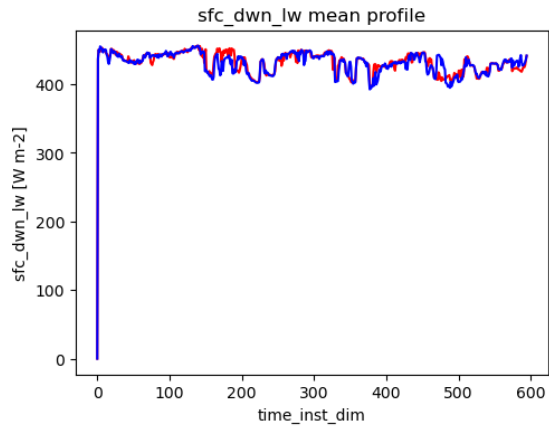
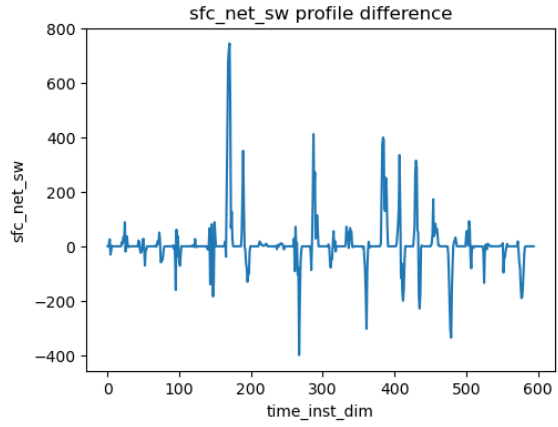
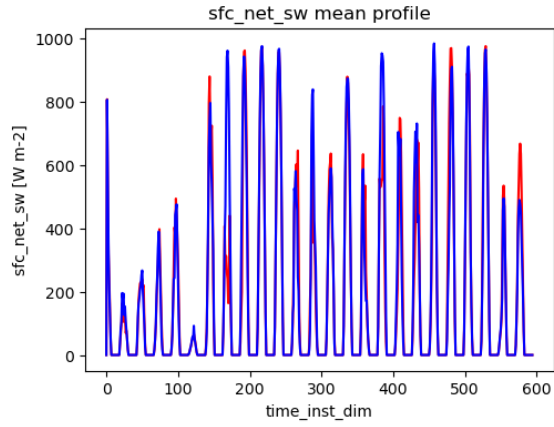


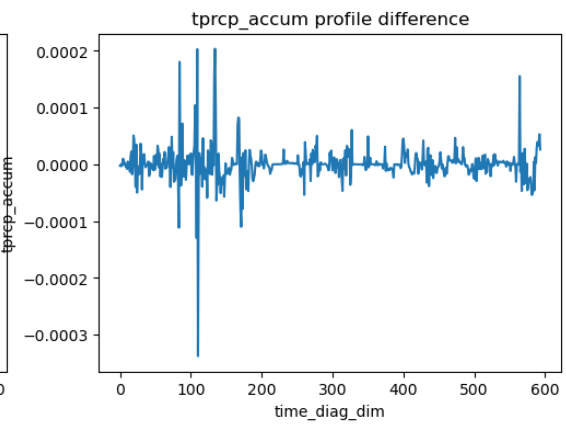
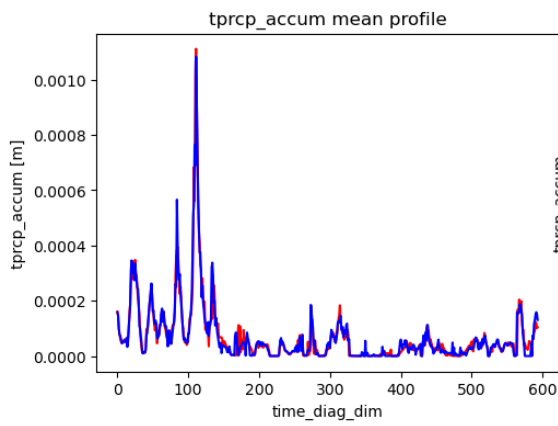
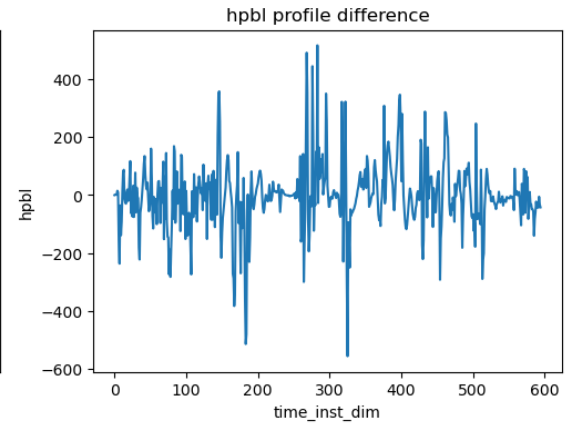
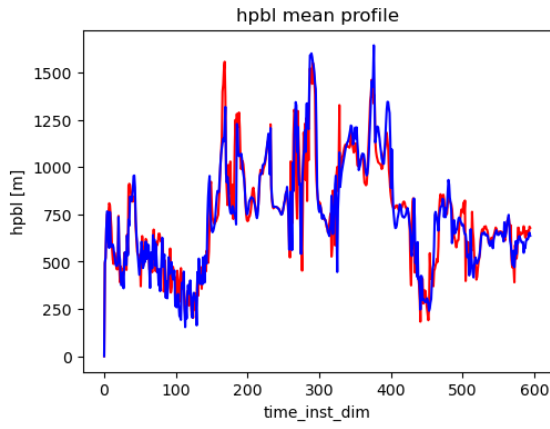
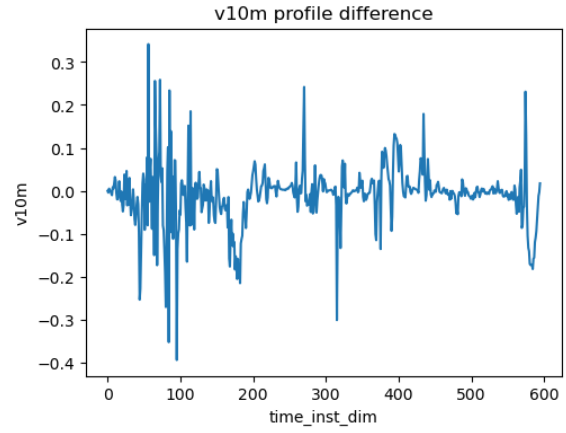
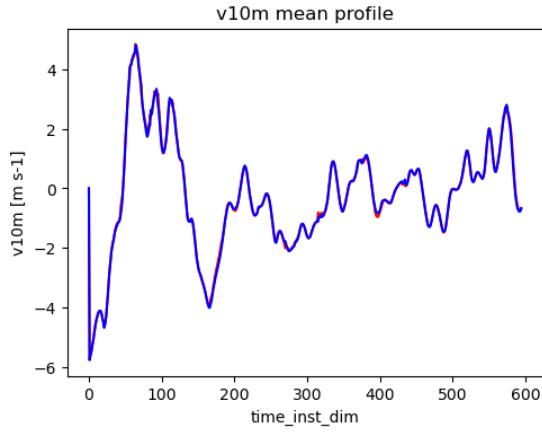


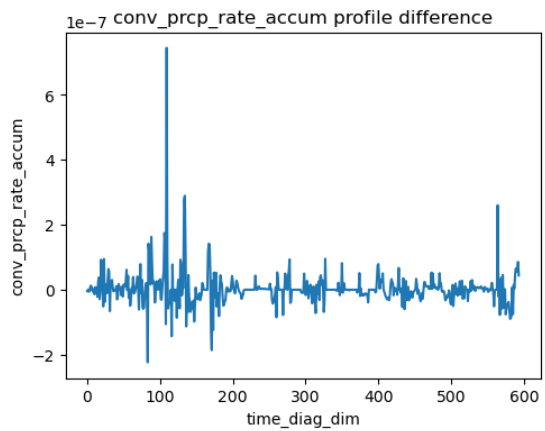
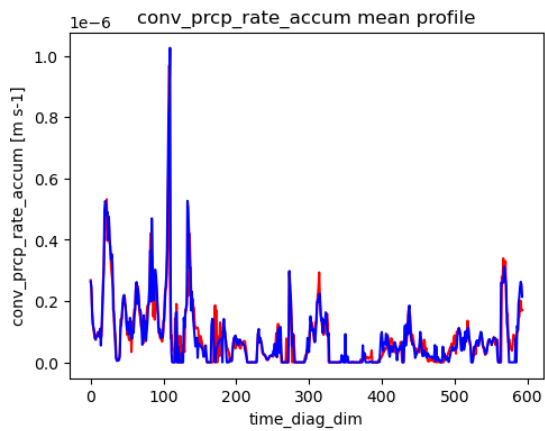
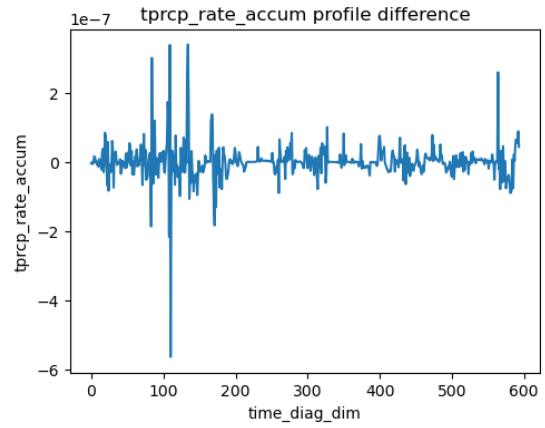
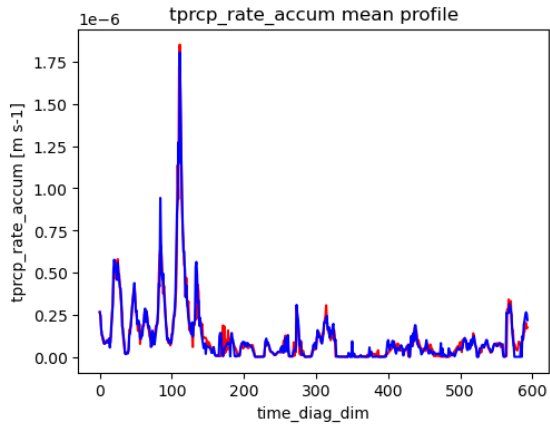
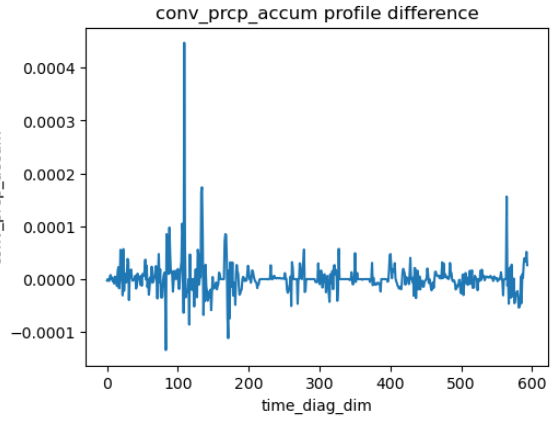
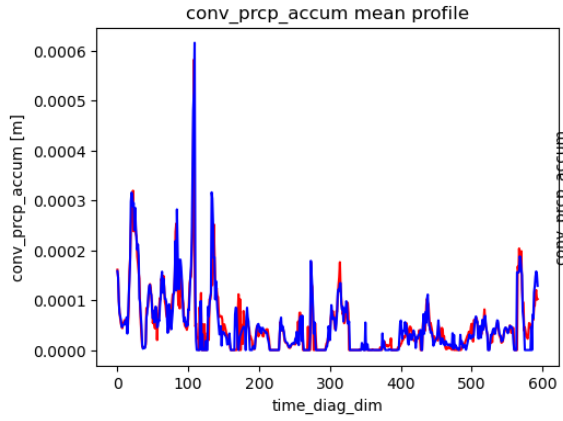


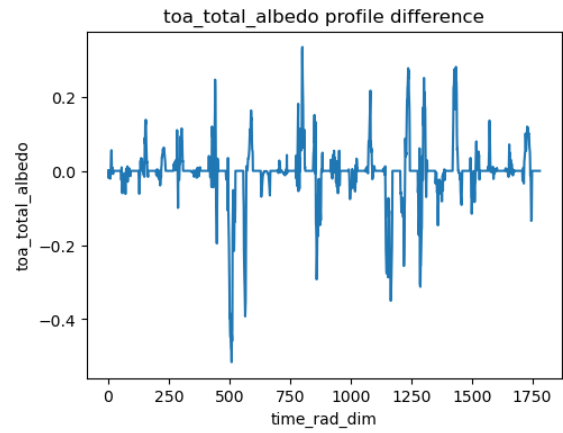
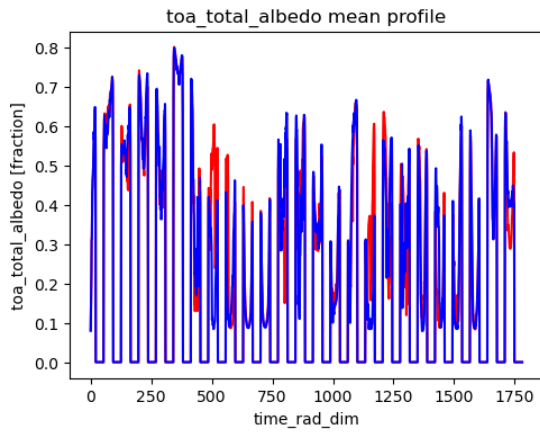
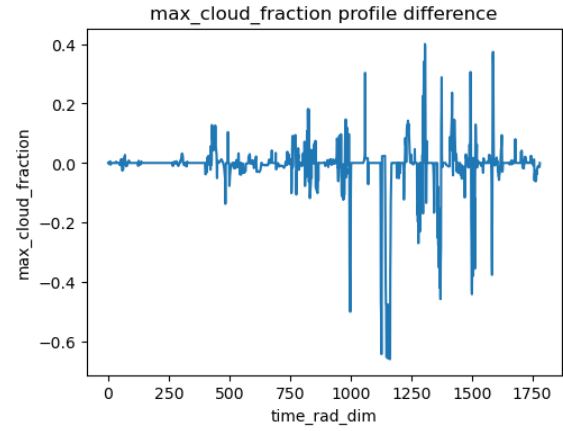
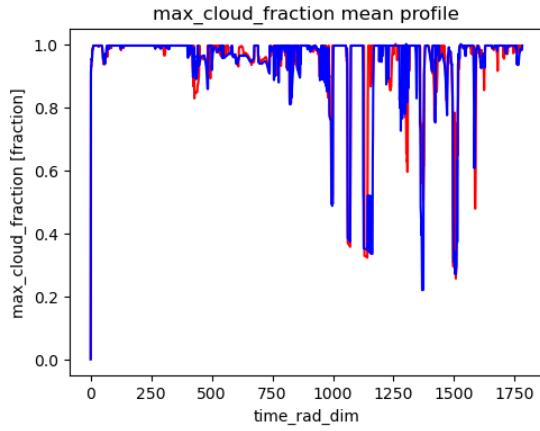












[]: