

single_vs_double_precision_output_t0

June 2, 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]: #mean_absolute_error = (abs(ds32[var] - ds64[var])[:, :, 0]) / ds32[var].count()
def print_error(var, max_re, std_err, mean_absolute_error):
    print("-----")
    print(var,
          ": max_re", max_re,
          ", std err =", std_err,
          ", mean absolute error", mean_absolute_error)
def calc_relative_error(ds32, ds64, var):
    if (len(ds32[var].shape) == 2):
        re = (abs(ds32[var] - ds64[var])[:, :]) / abs(ds64[var][:, :])
    else:
        re = (abs(ds32[var] - ds64[var])[:, :, 0]) / abs(ds64[var][:, :, 0])
    return re
def calc_relative_error_t0(ds32, ds64, var):
    if (len(ds32[var].shape) == 2):
        re = (abs(ds32[var] - ds64[var])[0, :]) / abs(ds64[var][0, :])
    else:
```

```

        re = (abs(ds32[var] - ds64[var])[0,:,:]) / abs(ds64[var][0,:,:])
    return re
def calc_std_error(re_std, count):
    return re_std / count
def calc_mean_absolute_error(ds32, ds64, var):
    if (len(ds32[var].shape) == 2):
        mean_absolute_error = np.sum(abs(ds32[var] - ds64[var])) / \
↳ds32[var].count()
    else:
        mean_absolute_error = np.sum(abs(ds32[var] - ds64[var])) / \
↳ds32[var].count()
    return mean_absolute_error
def calc_mean_absolute_error_t0(ds32, ds64, var):
    if (len(ds32[var].shape) == 2):
        mean_absolute_error = np.sum(abs(ds32[var] - ds64[var])[0,:]) / \
↳ds32[var].count()
    else:
        mean_absolute_error = np.sum(abs(ds32[var] - ds64[var])[0,:,:]) / \
↳ds32[var].count()
    return mean_absolute_error
def plot_relative_error(re, len_re):
    if (len_re == 2):
        fig, ax = plt.subplots(nrows=1, ncols=1, figsize=(6, 4))
        re.transpose().plot(ax=ax)
    else:
        fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(12, 4))
        re.transpose().plot(ax=axes[0], vmax=0.05)
        axes[0].set_title('Plot with vmax=0.05')
        re.transpose().plot(ax=axes[1], vmax=1.0)
        axes[1].set_title('Plot with vmax=1.0')
    plt.show()
def plot_relative_error_t0(re):
    fig, ax = plt.subplots(nrows=1, ncols=1, figsize=(4.5, 3))
    re.plot(ax=ax)
    plt.show()

```

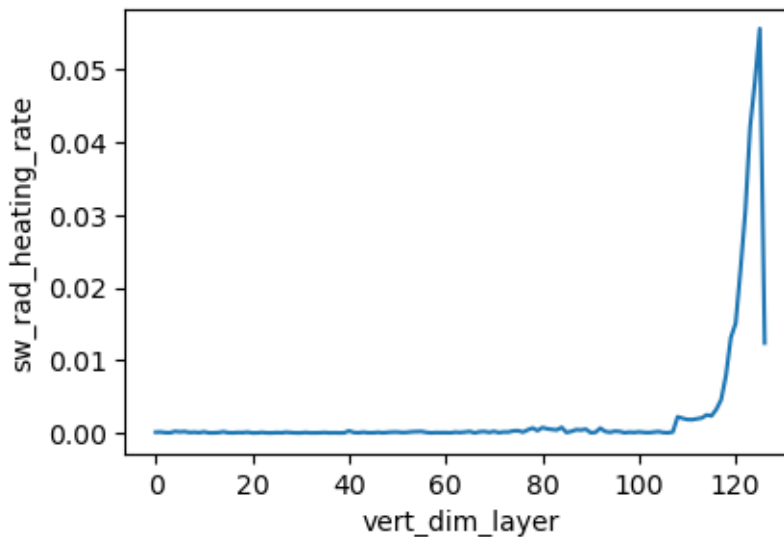
```

[ ]: # compare variables across full run
for var in ds32.variables:
    if (not ds32[var].identical(ds64[var])):
        relative_err = calc_relative_error(ds32, ds64, var)
        max_re = max(abs(relative_err.min().data), relative_err.max().data)
        if (max_re > .05):
            std_err = calc_std_error(relative_err.std(), relative_err.count())
            mean_absolute_err = calc_mean_absolute_error(ds32, ds64, var)
            print_error(var, max_re, std_err.data, mean_absolute_err.data)
            plot_relative_error(relative_err, len(ds32[var].shape))
            #sys.exit()

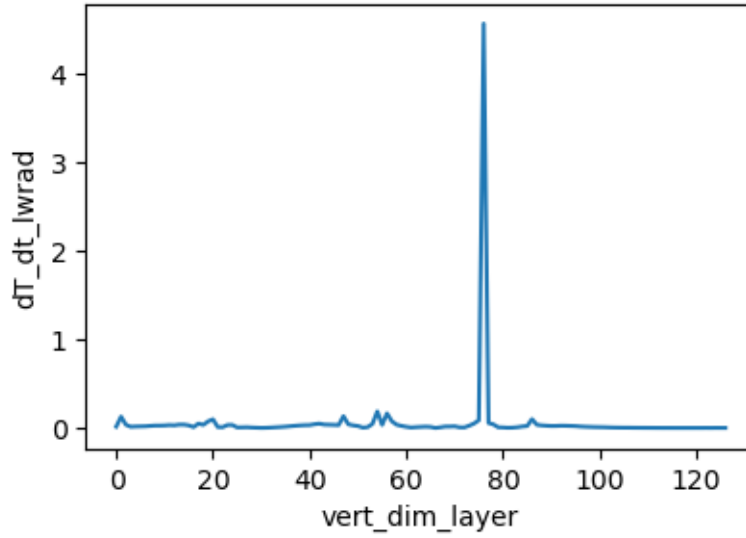
```

```
[5]: # compare variables at initial timestep
for var in ds32.variables:
    if (not ds32[var].identical(ds64[var])):
        relative_err = calc_relative_error_t0(ds32, ds64, var)
        max_re = max(abs(relative_err.min().data), relative_err.max().data)
        if (max_re > .05):
            std_err = calc_std_error(relative_err.std(), relative_err.count())
            mean_absolute_err = calc_mean_absolute_error_t0(ds32, ds64, var)
            print_error(var, max_re, std_err.data, mean_absolute_err.data)
            plot_relative_error_t0(relative_err)
            #sys.exit()
```

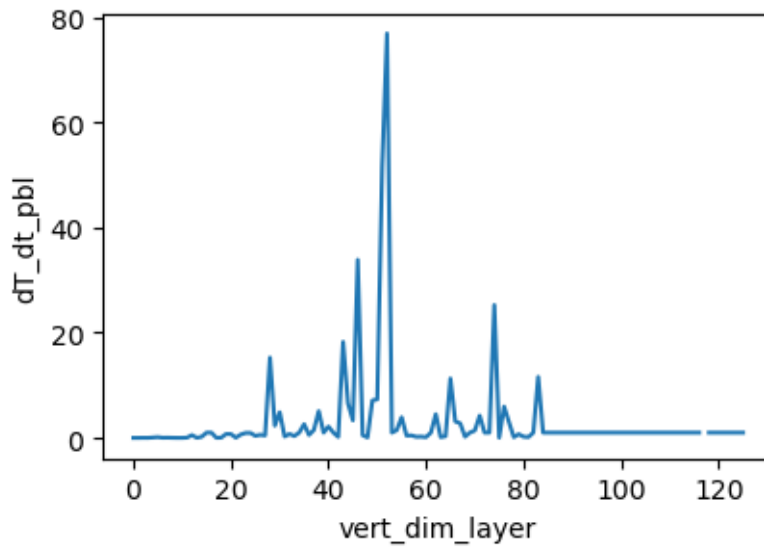
sw_rad_heating_rate : max_re 0.05562635511159897 , std err =
6.509930130064957e-05 , mean absolute error 2.1232059252188686e-10



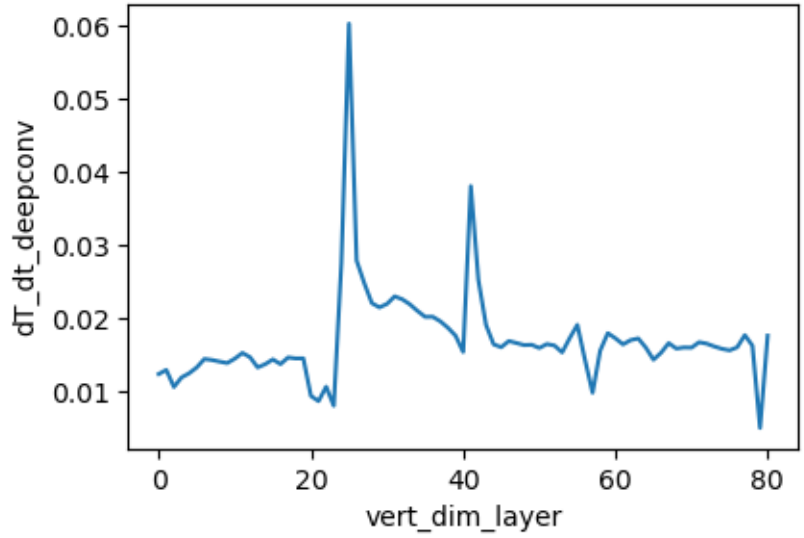
dT_dt_lwrad : max_re 4.558602809906006 , std err = 0.0031652600746455155 , mean
absolute error 6.380709468512078e-10



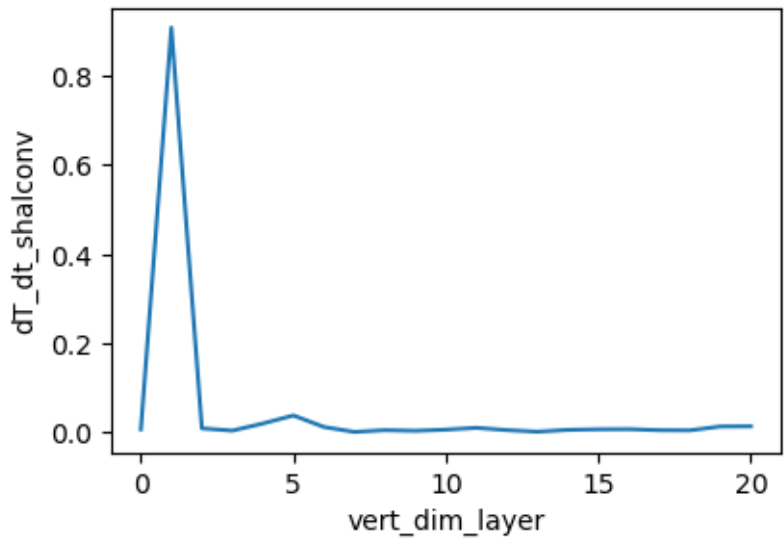
 dT_dt_pbl : max_re 76.87767028808594 , std err = 0.0730531005859375 , mean
 absolute error 8.42959976560591e-11



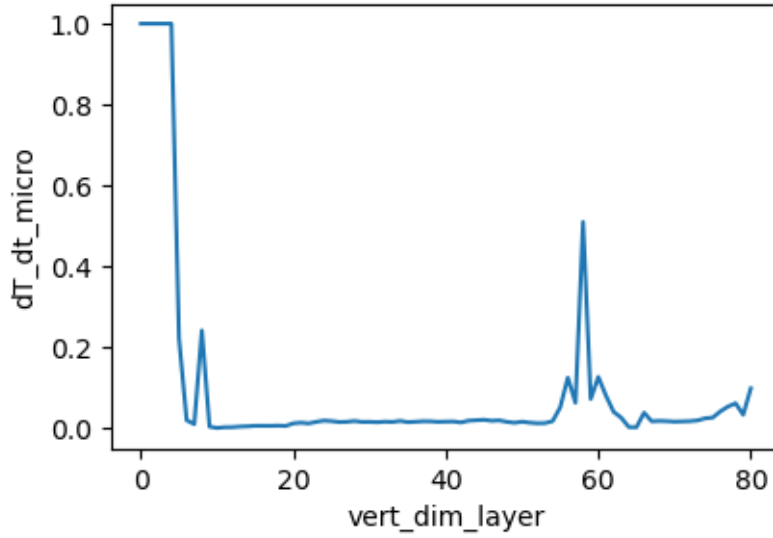
 $dT_dt_deepconv$: max_re 0.060232311487197876 , std err = 8.234146545514649e-05 ,
 mean absolute error 2.293594646174236e-09



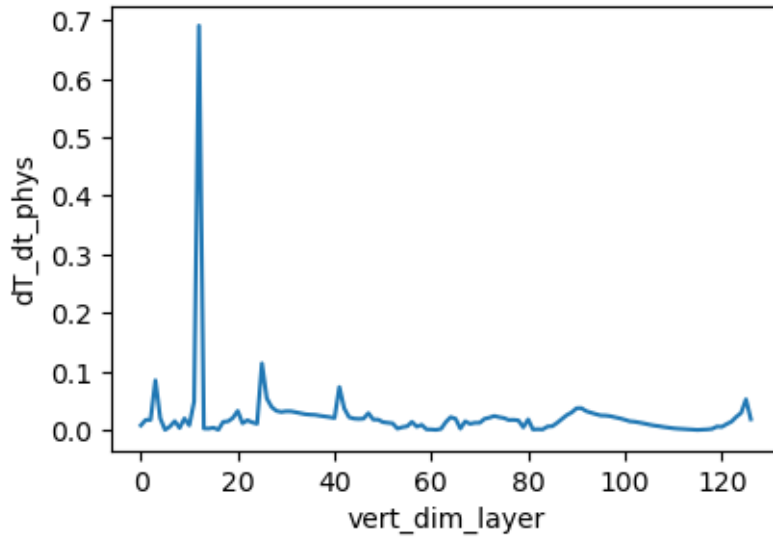
 dT_dt_shalconv : max_re 0.9087555408477783 , std err = 0.009141333046413604 ,
 mean absolute error 1.8514887869836624e-10



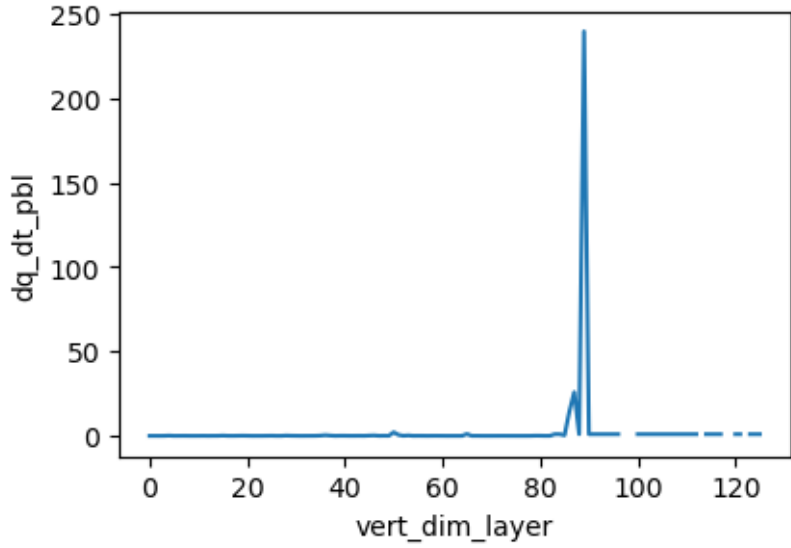
 dT_dt_micro : max_re 1.0 , std err = 0.00298064707973857 , mean absolute error
 6.881304572940394e-10



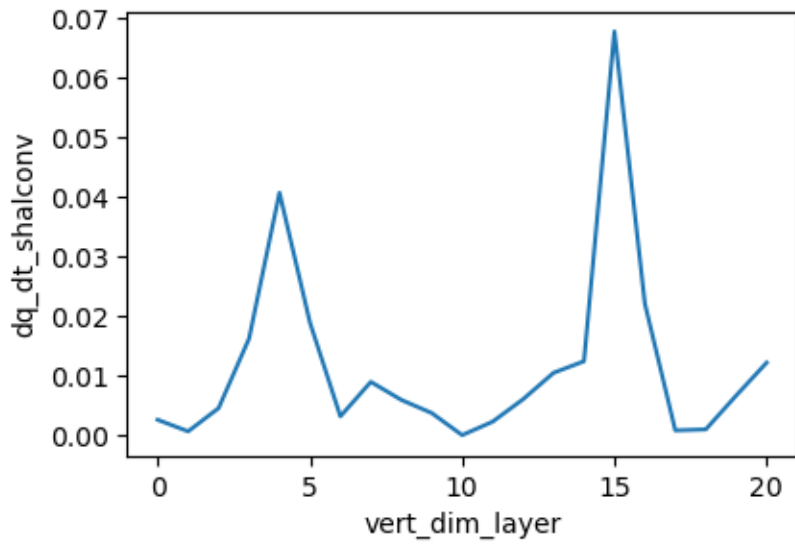
 dT_{dt_phys} : max_re 0.6906252503395081 , std err = 0.00048559029975275353 , mean absolute error 2.1940280880422787e-09



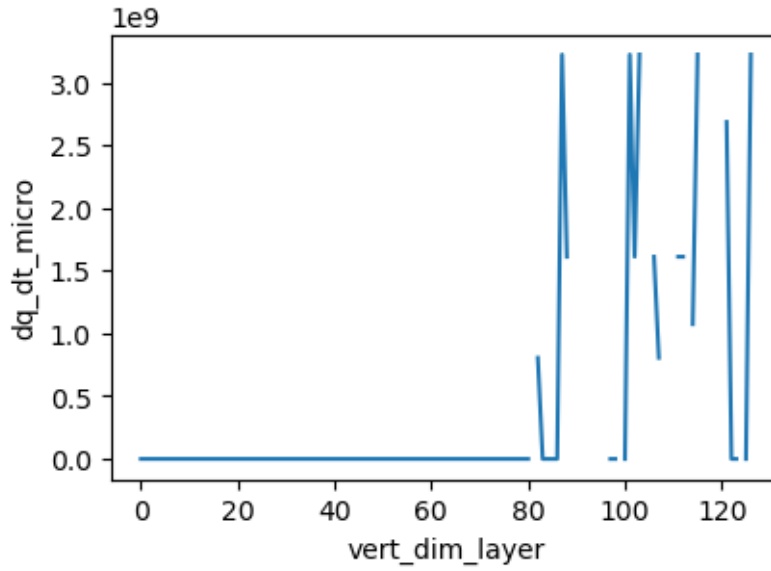
 dq_{dt_pbl} : max_re 239.51063537597656 , std err = 0.18229848543802898 , mean absolute error 2.3337351269606463e-13



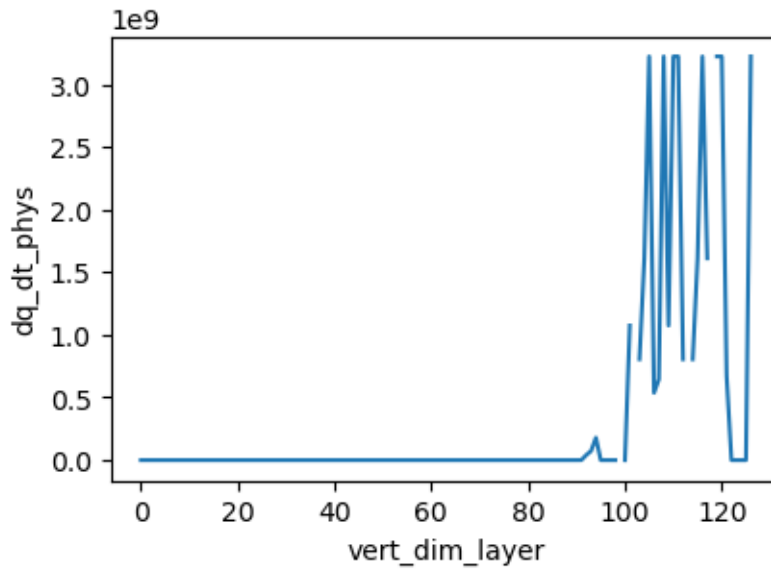
`dq_dt_shalconv` : max_re 0.06775882095098495 , std err = 0.00074244408658629371 ,
mean absolute error 2.910984696543385e-13



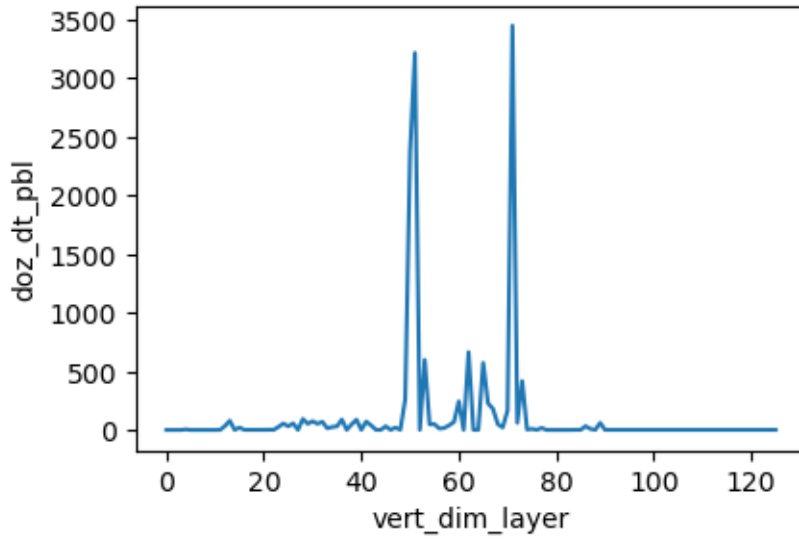
`dq_dt_micro` : max_re inf , std err = nan , mean absolute error
2.525139589695211e-13



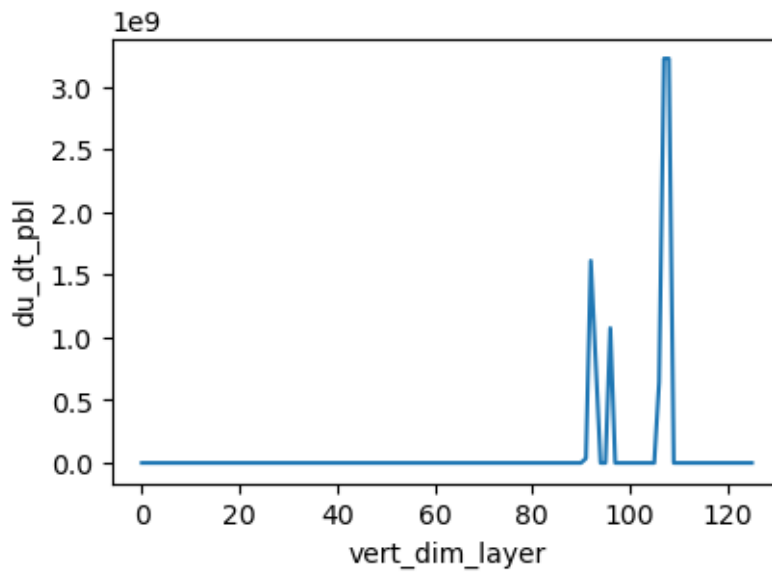
dq_dt_phys : max_re inf , std err = nan , mean absolute error
6.360278653158591e-13



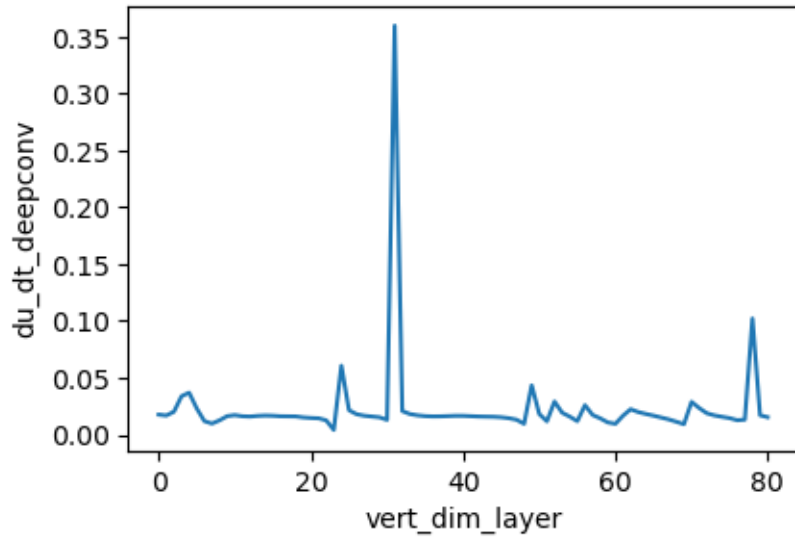
doz_dt_pbl : max_re 3443.158935546875 , std err = 3.732735043480283 , mean
absolute error 9.848343293310608e-21



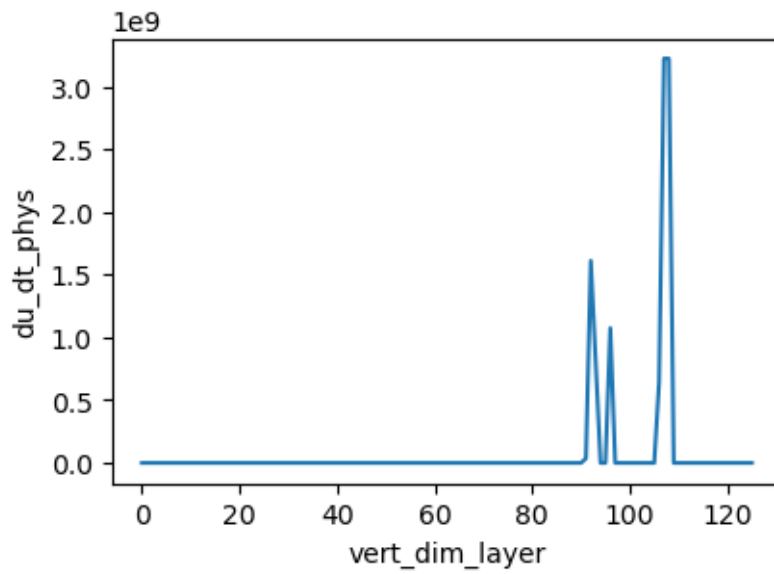
 du_dt_pbl : max_re 3221225472.0 , std err = 3511837.9682539683 , mean absolute error 4.892499422787199e-10



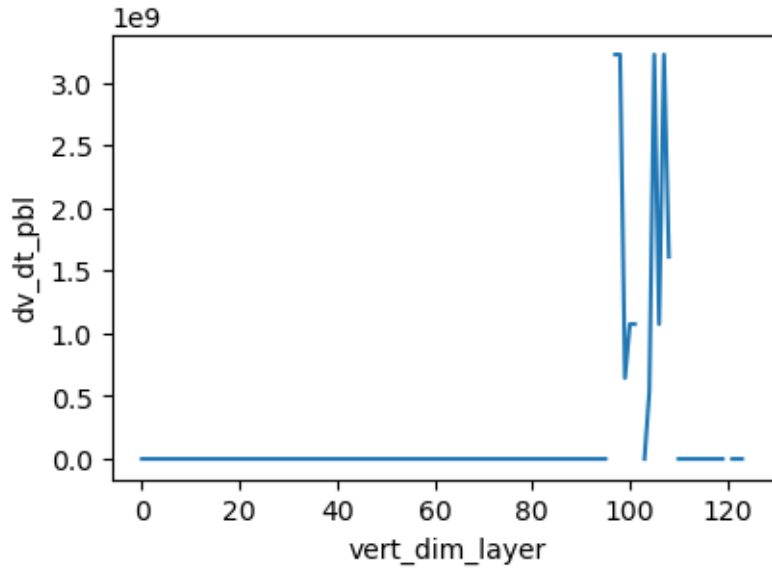
 du_dt_deepconv : max_re 0.35968348383903503 , std err = 0.00048793176258051836 , mean absolute error 2.118729382282069e-09



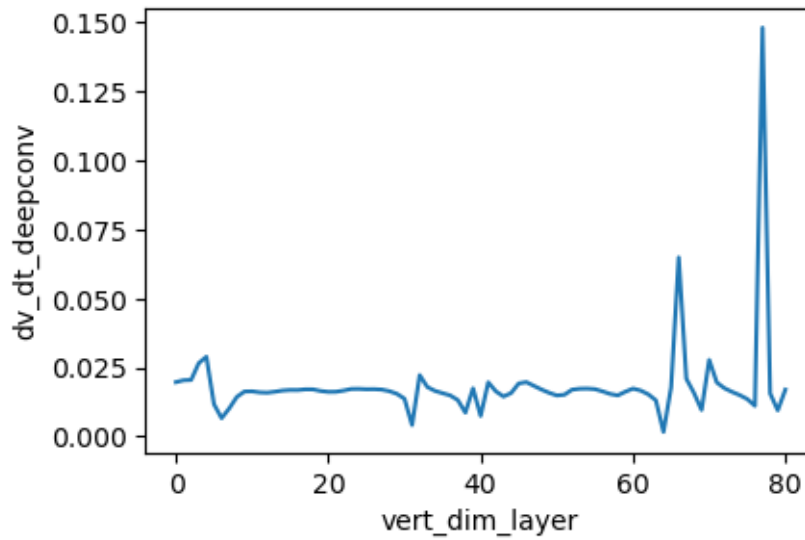
 du_dt_phys : max_re 3221225472.0 , std err = 3511837.9682539683 , mean absolute error 2.3112537785941844e-09



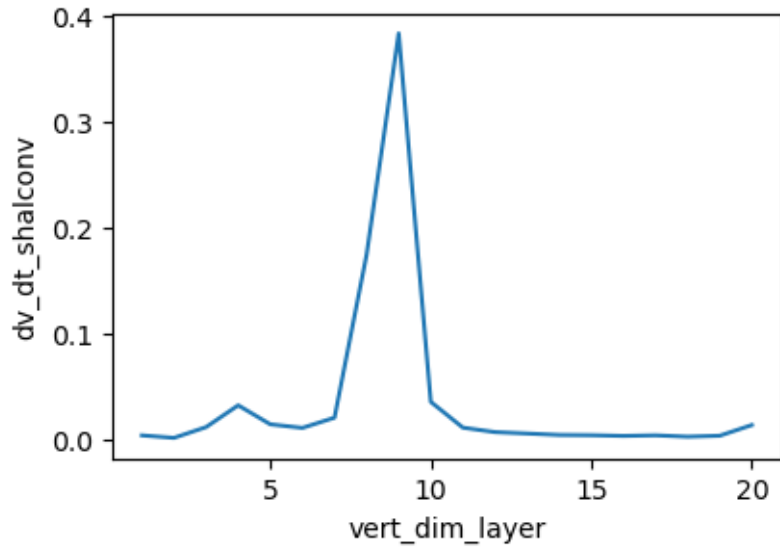
 dv_dt_pbl : max_re 3221225472.0 , std err = 5057291.636363637 , mean absolute error 1.0555585526253035e-10



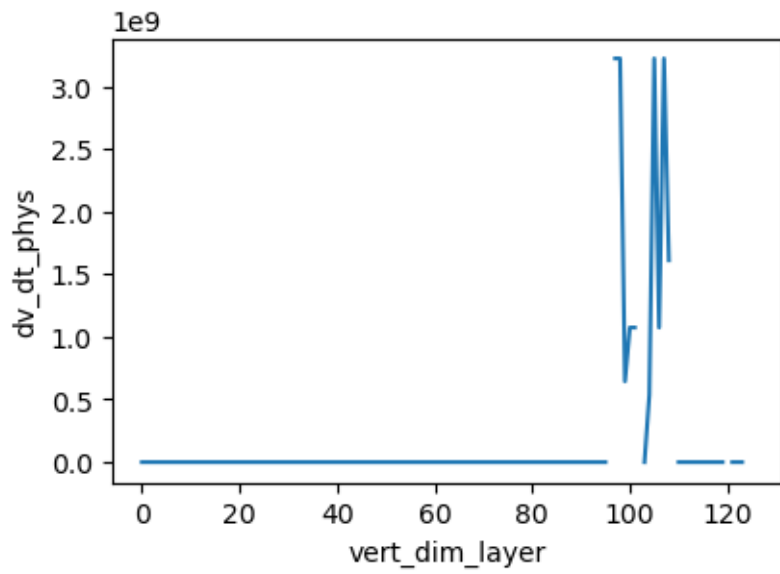
 dv_dt_deepconv : max_re 0.14795039594173431 , std err = 0.0001976865769168477 ,
 mean absolute error 1.6216502480394353e-09



 dv_dt_shalconv : max_re 0.3830767869949341 , std err = 0.004378025978803634 ,
 mean absolute error 1.0728652103362058e-10



`dv_dt_phys` : max_re 3221225472.0 , std err = 5057291.636363637 , mean absolute error 1.6576913054554965e-09



[]: