CMIP Greenhouse Gas (GHG) Concentration Historical Dataset

Zebedee Nicholls, Florence Bockting, Mika Pflüger

Oct 29, 2025

CONTENTS

CHAPTER

ONE

COMPARISON OF CMIP PHASES

1.1 Overview

Here we make some standalone plots that compare the historical concentrations over CMIP phases.

1.2 Data comparisons

Comparing the data from CMIP6 and CMIP7 shows minor changes (although doing this comparison requires a bit of care because of the changes in file formats).

```
fetch_and_load = partial(
    fetch_and_load_ghg_dataset,
    local_data_root_dir=local_data_root_dir,
    # index_node=KnownIndexNode.DKRZ,
    # cmip_era="CMIP6",
    # source_id="UoM-CMIP-1-2-0",
    index_node=KnownIndexNode.ORNL,
)
```

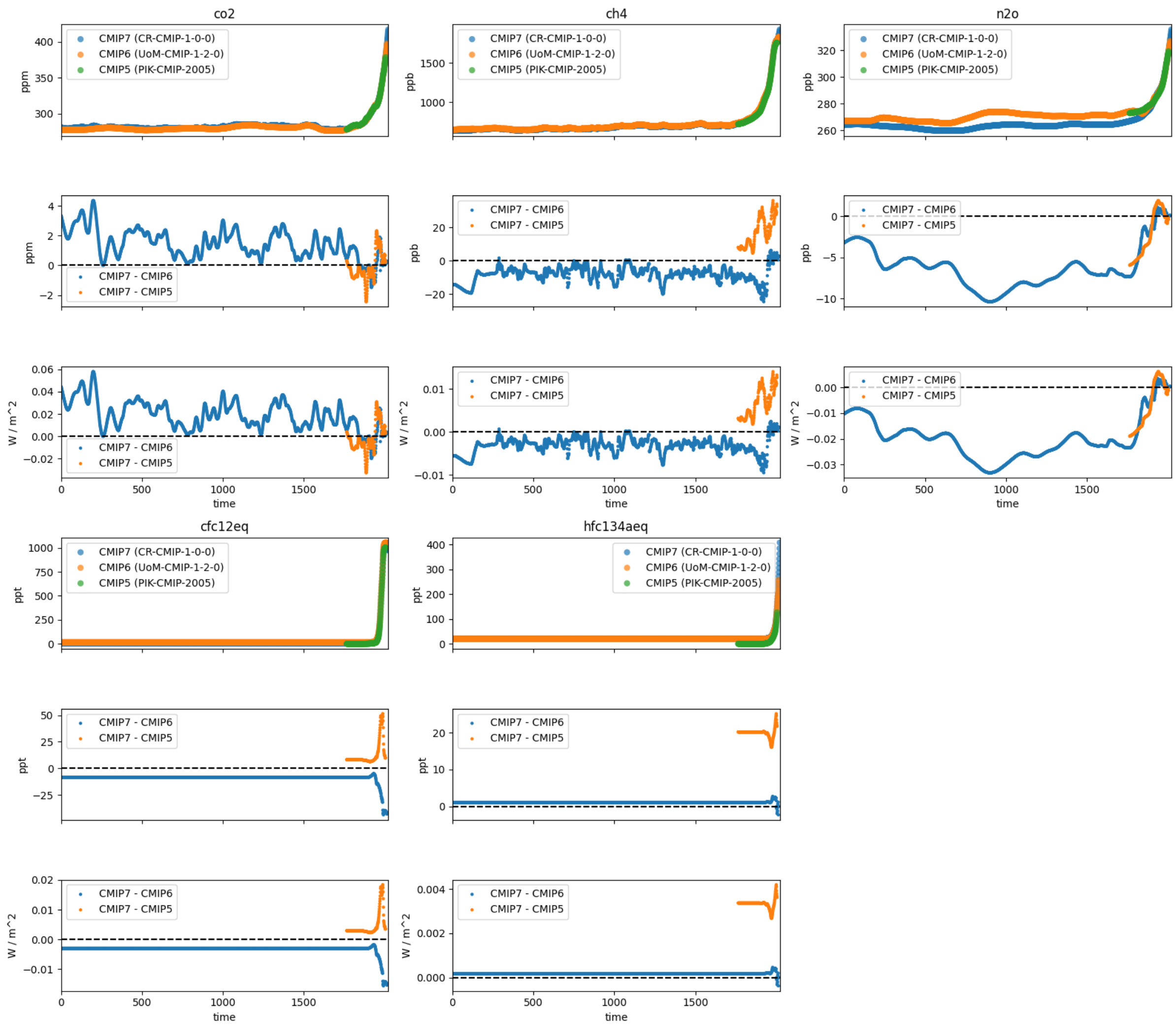
Values below come from Table 7.SM.7 of IPCC AR7 WG1 Ch. 7 Supplementary Material[4].

```
from openscm_units import unit_registry

Q = unit_registry.Quantity

RADIATIVE_EFFICIENCIES = {
    "co2": Q(1.33e-5, "W / m^2 / ppb"),
    "ch4": Q(3.88e-4, "W / m^2 / ppb"),
    "n2o": Q(3.2e-3, "W / m^2 / ppb"),
    "cfc12eq": Q(0.358, "W / m^2 / ppb"),
    "hfc134aeq": Q(0.167, "W / m^2 / ppb"),
}
```

[4] https://www.ipcc.ch/report/ar6/wg1/downloads/report/IPCC_AR6_WGI_Chapter07_SM.pdf

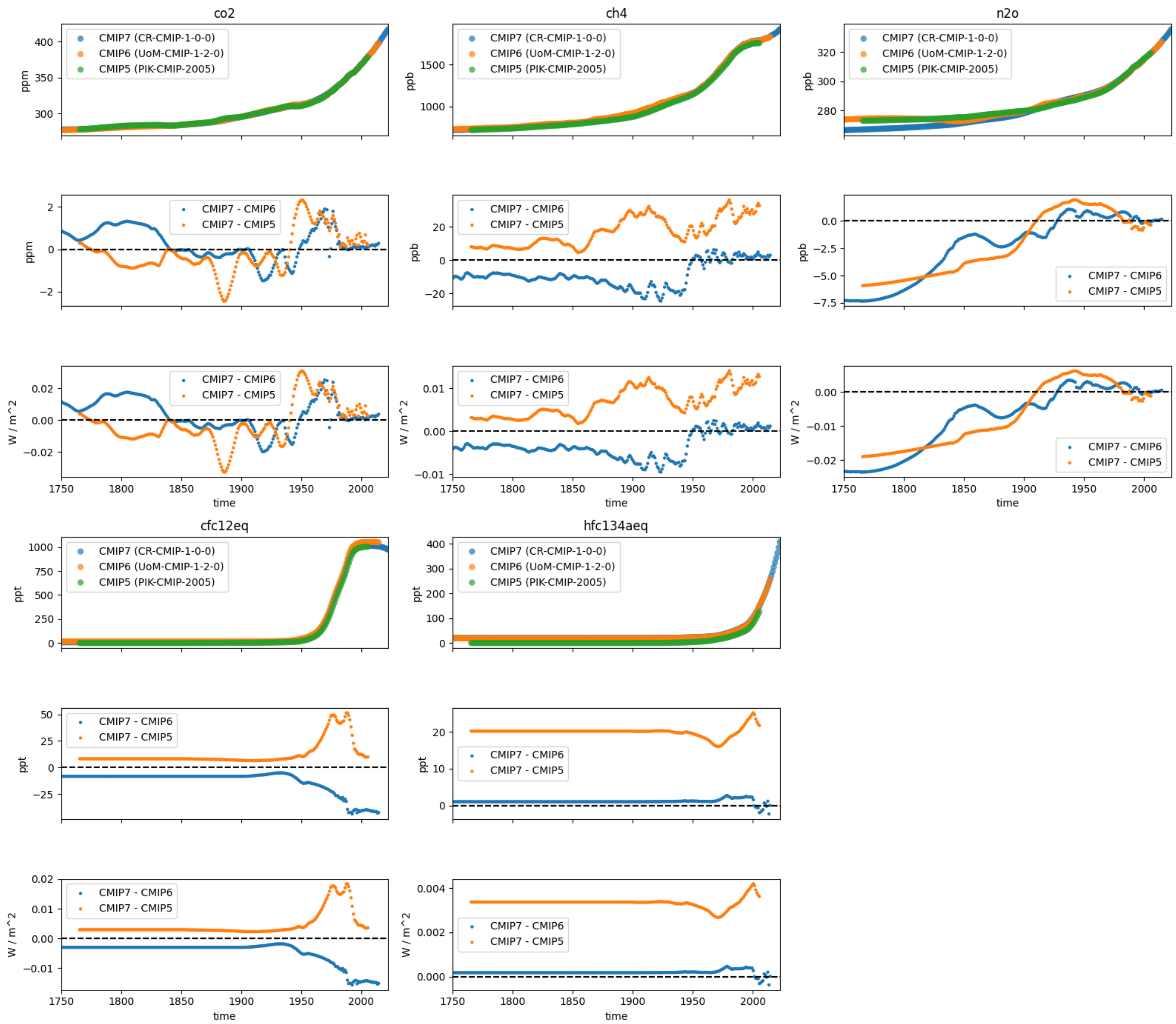
1.2.1 Global, annual-mean concentrations: Year 1 - 2022

```
plt.rcParams["axes.xmargin"] = 0
```



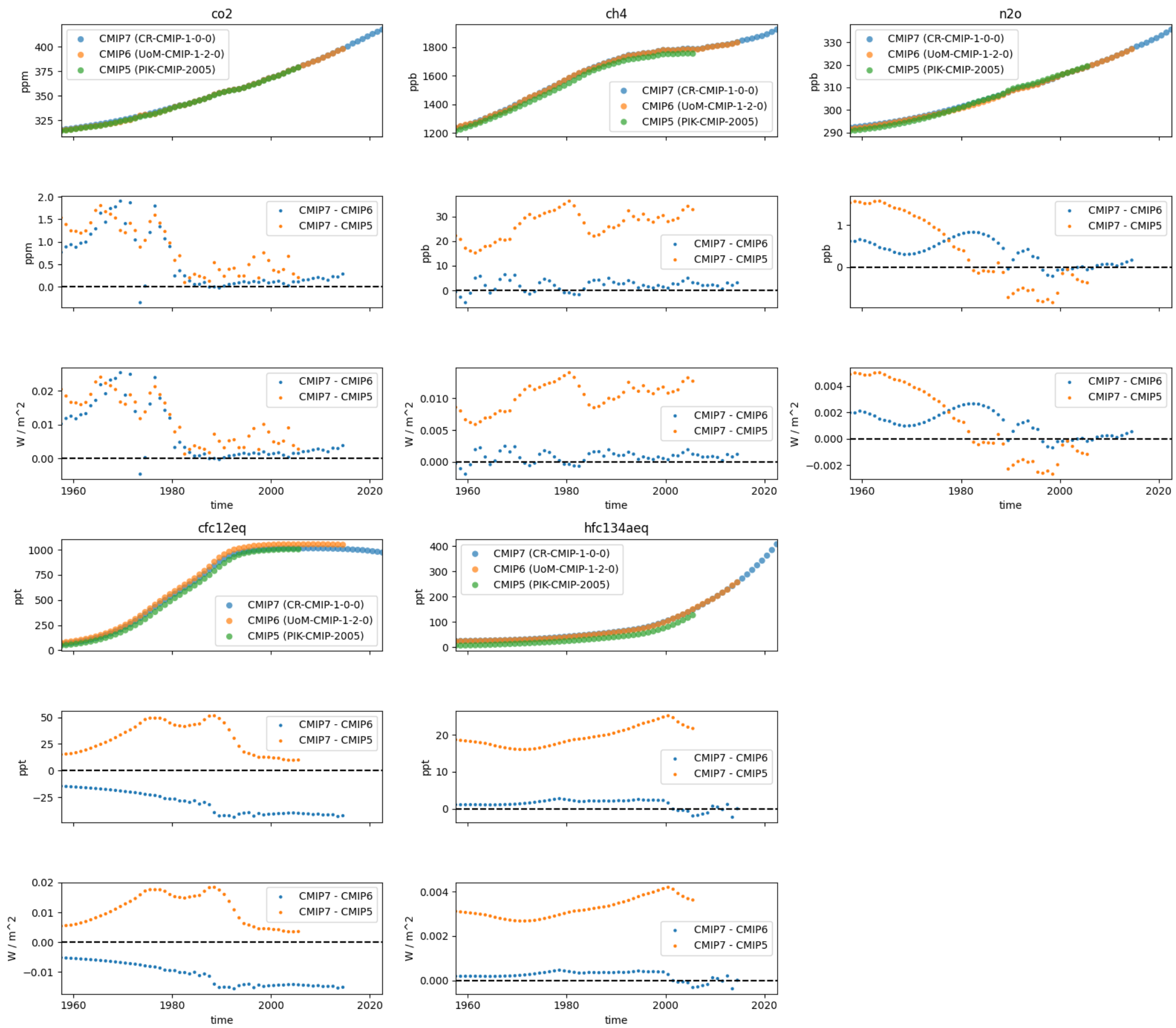
1.2.2 Global, annual-mean concentrations: Year 1750 - 2022

```
# TODO: copy https://github.com/climate-resource/CMIP6-vs-CMIP7-GHG-Concentrations/
↪blob/clean-up/notebooks/0101_demonstrate-cmip6-eq-issue.py
# into this repo to demonstrate the issue with the equivalent species
```



1.2.3 Global, annual-mean concentrations: Year 1957 - 2022

1957 is the start of the Scripps ground-based record. Before this, data is based on ice cores alone.



1.2.4 Global, monthly-mean concentrations: Year 1 - 2022

```
ds_gases_full_monthly_d = {}
for gas in gases_to_show:
    ds_gases_full_monthly_d[gas] = {}
    for source_id, cmip_era in (
        ("CR-CMIP-1-0-0", "CMIP7"),
        ("UoM-CMIP-1-2-0", "CMIP6"),
        # (None, "CMIP5"),
    ):
        query_kwargs = {
            "ghg": gas,
            "time_sampling": "mon",
            "grid": "gm",
            "target_mip": "CMIP",
            "source_id": source_id,
            "cmip_era": cmip_era,
            "engine": engine,
        }
```

(continues on next page)

(continued from previous page)

```
        ds = fetch_and_load(**query_kwargs)

        # Unify time axis days to simplify
        ds["time"] = [
            cftime.DatetimeProlepticGregorian(v.year, v.month, 15)
            for v in ds["time"].values
        ]

        # compute to avoid dask weirdness
        ds_gases_full_monthly_d[gas][cmip_era] = ds.compute()
```

```
Datasets: 0it [00:00, ?it/s]
```

```
Datasets: 0it [00:00, ?it/s]
```

```
Datasets: 0it [00:00, ?it/s]
```

```
Datasets: 0it [00:00, ?it/s]
```

```
Datasets: 0it [00:00, ?it/s]
```

```
Datasets: 0it [00:00, ?it/s]
```

```
Datasets: 0it [00:00, ?it/s]
```

```
Datasets: 0it [00:00, ?it/s]
```

```
Datasets: 0it [00:00, ?it/s]
```

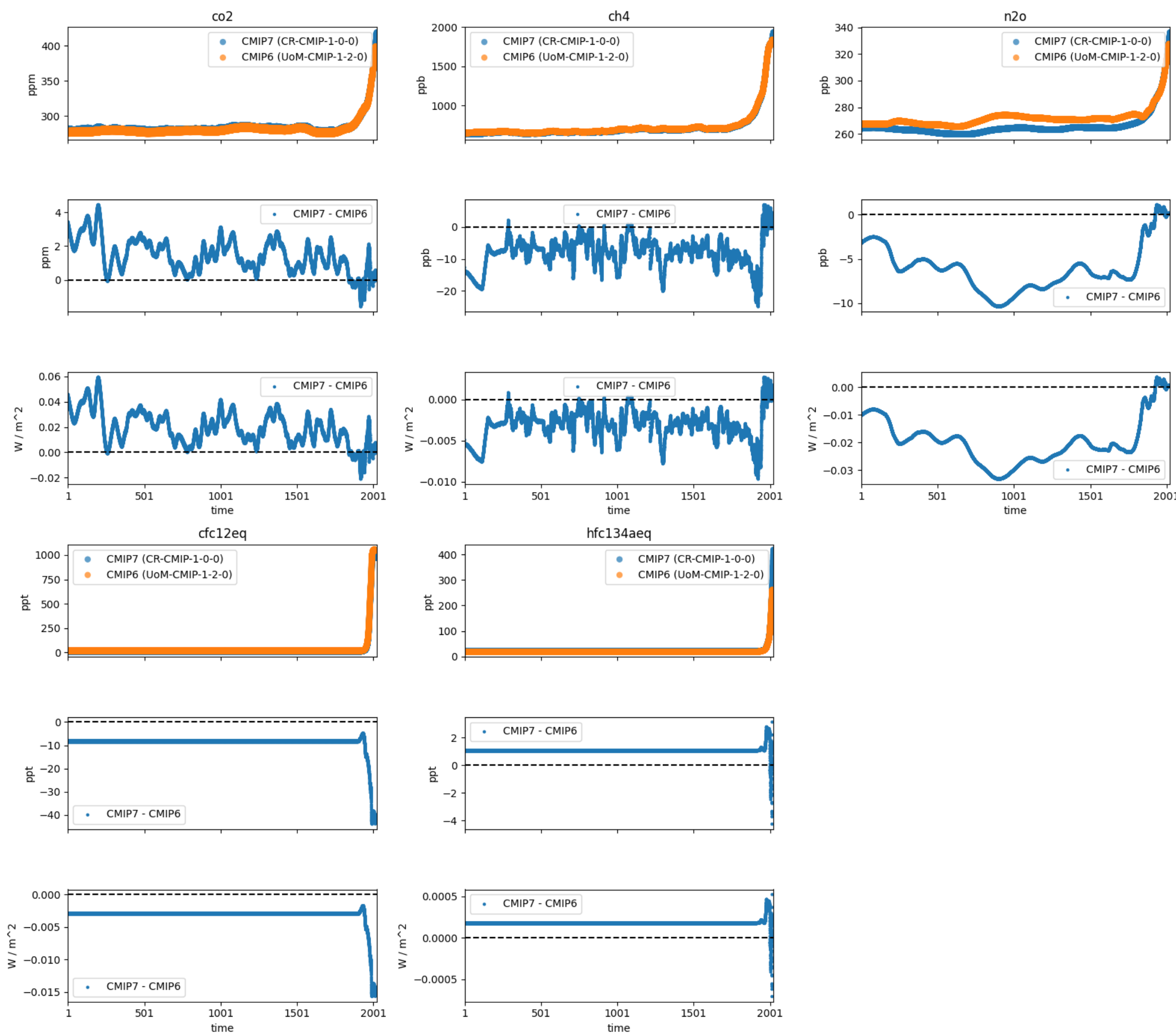
```
Datasets: 0it [00:00, ?it/s]
```

```
fig, axes_d = get_default_delta_mosaic()
axes_d = remove_empty_axes(axes_d)

# min_year = 1957
min_year = 1990
min_year = 1
# min_year = 1750
plot_overview_and_deltas(
    sel_times(ds_gases_full_monthly_d, lambda x: x.dt.year >= min_year),
    axes_d,
)
for ax in axes_d.values():
    xticks = [
        cftime.DatetimeProlepticGregorian(y, 1, 1)
        # for y in np.arange(1750, 2050, 50)
        # for y in np.arange(1750, 1760, 1)
        for y in np.arange(1, 2050, 500)
        # for y in np.arange(1, 20, 1)
    ]
    ax.set_xticks(xticks)
    # ax.set_xlim(xticks[0], xticks[-1])
```

(continues on next page)

(continued from previous page)

```
    ax.set_xticklabels([v.year for v in xticks])

plt.tight_layout()
plt.show()
```



1.2.5 Latitudinally-resolved, monthly-mean concentrations: Year 1 - 2022

```
gases_to_show = ["co2", "ch4"]
ds_gases_full_monthly_lat_d = {}
for gas in gases_to_show:
    ds_gases_full_monthly_lat_d[gas] = {}
    for source_id, cmip_era in (
        ("CR-CMIP-1-0-0", "CMIP7"),
        ("UoM-CMIP-1-2-0", "CMIP6"),
        # (None, "CMIP5"),
    ):
        query_kwargs = {
            "ghg": gas,
```

(continues on next page)

(continued from previous page)

```
            "time_sampling": "mon",
            "grid": "gnz",
            "target_mip": "CMIP",
            "source_id": source_id,
            "cmip_era": cmip_era,
            "engine": engine,
        }
        ds = fetch_and_load(**query_kwargs)

        # Unify time axis days to simplify
        ds["time"] = [
            cftime.DatetimeProlepticGregorian(v.year, v.month, 15)
            for v in ds["time"].values
        ]

        # compute to avoid dask weirdness
        ds_gases_full_monthly_lat_d[gas][cmip_era] = ds.compute()
```

```
Datasets: 0it [00:00, ?it/s]
```

```
Datasets: 0it [00:00, ?it/s]
```

```
Datasets: 0it [00:00, ?it/s]
```

```
Datasets: 0it [00:00, ?it/s]
```

```
def plot_lat_selection(
    gas: str,
    ds_d: dict[str, dict[str, xr.Dataset]],
    ax: matplotlib.axes.Axes,
    ax_delta: matplotlib.axes.Axes,
    ax_delta_re: matplotlib.axes.Axes,
) -> None:
    """
    Plot selection for a latitude-specific dataset
    """
    target_unit_conc = ds_d[gas]["CMIP7"][gas].attrs["units"]
    target_unit_re = "W / m^2"

    for cmip_era, ds in ds_d[gas].items():
        label = f"{cmip_era} ({ds.attrs['source_id']})"
        tmp = ds[gas].copy()
        tmp.values = Q(tmp.values, tmp.attrs["units"]).to(target_unit_conc).m
        ds[gas].plot.scatter(ax=ax, label=label, alpha=0.7, edgecolors="none")

    ax.legend()
    ax.set_title(
        f"lat: {float(lat)}",
        # fontsize="small",
    )
    ax.xaxis.set_tick_params(labelbottom=False)
    ax.set_ylabel(target_unit_conc)
    ax.set_xlabel(None)

    da_cmip7 = ds_d[gas]["CMIP7"][gas]
```

(continues on next page)

(continued from previous page)

```
    for cmip_era, ds in ds_d[gas].items():
        if cmip_era == "CMIP7":
            continue

        da_other = ds_d[gas][cmip_era][gas]
        overlapping_times = np.intersect1d(da_other["time"], da_cmip7["time"])

        da_cmip7_st = da_cmip7.sel(time=overlapping_times)
        da_other_st = da_other.sel(time=overlapping_times)

        delta = da_cmip7_st.copy()
        tmp = Q(da_cmip7_st.values, da_cmip7_st.attrs["units"]) - Q(
            da_other_st.values, da_other_st.attrs["units"]
        )
        delta.values = tmp.to(target_unit_conc).m

        delta.plot.scatter(
            ax=ax_delta,
            label=f"CMIP7 - {cmip_era}",
            edgecolors="none",
            s=10,
        )
        ax_delta.axhline(0.0, color="k", linestyle="--")
        ax_delta.legend()

        ax_delta.xaxis.set_tick_params(labelbottom=False)
        ax_delta.set_ylabel(target_unit_conc)
        ax_delta.set_xlabel(None)
        ax_delta.set_title(None)

        tmp = RADIATIVE_EFFICIENCIES[gas] * Q(delta.values, delta.attrs["units"])
        delta_re = delta.copy()
        delta_re.values = tmp.to(target_unit_re).m
        delta_re.attrs["units"] = target_unit_re

        delta_re.plot.scatter(
            ax=ax_delta_re,
            label=f"CMIP7 - {cmip_era}",
            edgecolors="none",
            s=10,
        )
        ax_delta_re.axhline(0.0, color="k", linestyle="--")

        ax_delta_re.xaxis.set_tick_params(labelbottom=True)
        ax_delta_re.set_ylabel(target_unit_re)
        ax_delta_re.legend()
        ax_delta_re.set_title(None)
```

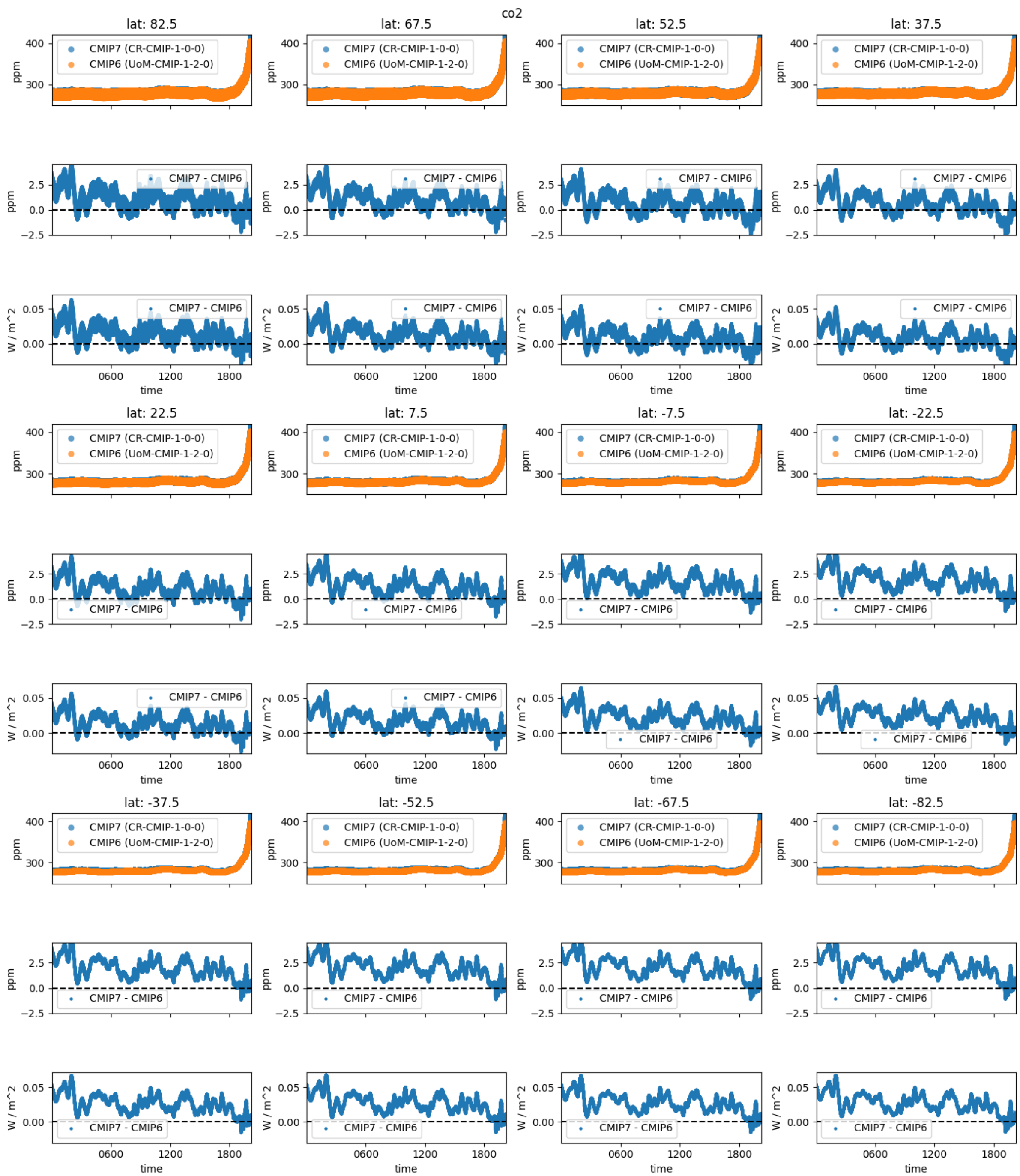
```
gas = "co2"
# gas = "ch4"
min_year = 1
# min_year = 1750
# min_year = 1850
# min_year = 2000
sel_times_func = lambda x: (x.dt.year >= min_year)  # noqa: E731
```

(continues on next page)

(continued from previous page)

```
# sel_times_func = lambda x: (x.dt.year >= min_year) & (x.dt.year <= min_year + 2)

ncols = 4
fig, axes = plt.subplots(ncols=ncols, nrows=9, figsize=(14, 16), sharex=True)
ax_flat = axes.flatten()

for i, lat in tqdm.auto.tqdm(
    enumerate(ds_gases_full_monthly_lat_d[gas]["CMIP7"]["lat"][::-1]), leave=False
):
    ax_idx = i % ncols + 3 * ncols * (i // ncols)
    # print(ax_idx)
    ax = ax_flat[ax_idx]
    ax_delta = ax_flat[ax_idx + ncols]
    ax_delta_re = ax_flat[ax_idx + 2 * ncols]

    plot_lat_selection(
        gas=gas,
        ds_d=sel_times(
            sel_lat(ds_gases_full_monthly_lat_d, lambda x: x == lat),
            sel_times_func,
        ),
        ax=ax,
        ax_delta=ax_delta,
        ax_delta_re=ax_delta_re,
    )
    # ax_flat[ax_idx].legend().remove()
    if gas == "co2":
        ax.set_ylim([250, 420])
        ax_delta.set_ylim([-2.5, 4.5])
        ax_delta_re.set_ylim([-0.03, 0.071])
    elif gas == "ch4":
        ax.set_ylim([600, 1900])
        ax_delta.set_ylim([-70, 35])
        ax_delta_re.set_ylim([-0.028, 0.02])
    # # break

plt.tight_layout()
# plt.savefig(f"{gas}_lat-monthly.png")
plt.suptitle(gas, y=1.0)
plt.show()
```

```
0it [00:00, ?it/s]
```


co2
lat: 82.5
lat: 67.5
lat: 52.5
lat: 37.5
lat: 22.5
lat: 7.5
lat: -7.5
lat: -22.5
lat: -37.5
lat: -52.5
lat: -67.5
lat: -82.5
CMIP7 (CR-CMIP-1-0-0)
CMIP6 (UoM-CMIP-1-2-0)
CMIP7 - CMIP6
ppm
W / m^2
time
0600
1200
1800