CMIP Greenhouse Gas (GHG) Concentration Scenarios Dataset

Zebedee Nicholls, Florence Bockting, Mika Pflüger

Oct 29, 2025

CONTENTS

CHAPTER

ONE

DATA DESCRIPTION AND USER GUIDE

1.1 Overview

Here we provide a short description of the draft scenario dataset and a guide for users. This is intended to provide a short introduction for users of the data. The full details of the dataset's construction and evaluation against other data sources will be provided in the full manuscript which is being prepared.

1.2 Dataset construction

The dataset is constructed following a similar methodology to Meinshausen *et al.* [1]. The full method will be described in a forthcoming paper. In brief, the method is:

1. **retrieve concentrations** of GHG gases
    1. for gases covered under the Montreal Protocol (TODO REF) and whose concentration evolution is already specified in WMO 2022 (TODO, ref), we simply use the WMO 2022 concentrations
    2. for all other gases, we
        1. start with harmonised, complete emissions sets for each scenario of interest
        2. run MAGICC (TODO REF) to translate these emissions into global-mean concentrations
            1. MAGICC is run in the same configuration which was used in AR6 (as described/evaluated in Cross-Chapter Box 7.1[1]). This represents our best estimate, in line with the last IPCC report, of the concentrations that result from the emissions. Note, given that CMIP7 models have not yet been run, this will, almost by definition, produce different concentrations than those outputed by the ESMs run in emissions-driven mode. We nonetheless look forward to reading lots of papers saying that "CMIP7 input concentrations are biased up/down/whatever".
        3. harmonise the global-mean concentrations to the historical concentrations using gradient-aware harmonisation[2] to ensure a smooth transition in both the absolute values and the gradient, improving on the CMIP6 data which had an abrupt jump in the gradient (most notable for methane, further details below)
2. **seasonality and latitudinal gradients**: We use the same statistical models for seasonality and latitudinal gradients as were used over the historical period. These are then applied to the future, based on the same input drivers.
    - one exception is CO2, which uses GPP as the input driver rather than a regression on a combined global-mean surface temperature and CO2 concentration metric (as is used in the product that covers history)
    - arguably we should harmonise the inputs to these models too. We don't do this step as such a harmonisation would only provide second-order corrections.

[1] https://www.ipcc.ch/report/ar6/wg1/chapter/chapter-7/
[2] https://github.com/climate-resource/gradient-aware-harmonisation

3. **combine the global-means, seasonality and latitudinal gradients** to produce our complete, gridded projections
4. **calculate lower resolution products** from the gridded product
5. **release** all the data

1.3 Finding and accessing the data

1.3.1 ESGF

The Earth System Grid Federation (ESGF, TODO REF) provides access to a range of climate data.

The scenario data of interest here, which is a draft dataset can be found under "MIP era" `CMIP6Plus` (for draft datasets), "institution ID" `CR` and "source version" `0.1.0` (also under "source IDs" of the form `CR-*-0-1-0`, although this is less useful as the ESGF search API does not appear to support glob/star expressions for specific facet searches).

It is possible to filter searches on ESGF via the user interface. Searches can often be encoded in URLs[3] too (although these URLs sometimes move, so we make no guarantee that this link will always be live).

These datasets are a draft only. The final datasets will maintain the same form; however, the numbers are currently being finalised and the final names have been changed since the draft datasets were published, so please take care to treat the values shown here as drafts only.

To download the data, we recommend accessing it directly via the ESGF user interfaces via links as discussed above. Alternately, there are tools dedicated to accessing ESGF data, with two prominent examples being:

1. esgpull: https://esgf.github.io/esgf-download
2. intake-esgf: https://intake-esgf.readthedocs.io

Please see these tools' docs for usage instructions.

1.3.2 Zenodo

While it aims to be, the ESGF is technically not a permanent archive and does not issue DOIs. In order to provide more reliable, citable access to the data, we will also provide the final scenario datasets on Zenodo (TODO REF, although we have not done this step for the draft datasets). When ready, we will update this guide to use the final scenario data and include the zenodo link to the source code and input data used to process it.

1.4 Data description

1.4.1 Format

The data is provided in netCDF format [TODO citation]. This self-describing format allows the data to be placed in the same file as metadata (in the so-called "file header"). The scenario datasets are only 78 years long, so each scenario-specific dataset is provided in a single file (unlike the historical dataset, which is split into multiple files).

[3] An example URL: https://esgf-node.ornl.gov/search?project=input4MIPs&activeFacets=%7B%22source_version%22%3A%220.1.0%22%2C%22institution_id%22%3A%22CR%22%2C%22mip_era%22%3A%22CMIP6Plus%22%7D

1.4.2 Scenario information

Determining the scenario to which each dataset applies is not trivial (there was discussion about how to make this more trivial, but ultimately backwards-compatibility was prioritised).

Each dataset is given a unique source ID. This source ID appears both in the global attributes of the file as well as in the filename (as the 5th underscore separated element). The concept of a source ID is a bit of a perculiar one to CMIP forcings data. In practice, it is simply a unique identifier for a collection of datasets (and it's best not to read more than that into it).

As a result of the way that scenario data is handled in CMIP/input4MIPs, the scenario name appears as part of the source ID, rather than as a standalone attribute/identifier. This means that its value must be extracted from the other parts of the source ID. In general, this is not a trivial problem (users of forcings data more generally are directed to input4mips-cvs.readthedocs.io/en/latest/extracting-scenario-from-source-id). For the greenhouse gas concentrations, the scenario identifier is simply the second hyphen-separated element of the source ID. For example, for the source ID `CR-ml-0-1-0`, the scenario identifier is `ml`. A Python function for doing this extraction is below.

```
def extract_scenario_id(source_id: str) -> str:
    """
    Extract scenario ID from a GHG concentration source ID

    Parameters
    ----------
    source_id
        Source ID from which to extract the scenario ID

    Returns
    -------
    :
        Scenario ID
    """
    return source_id.split("-")[1]


print(f"{extract_scenario_id('CR-ml-0-1-0')=}")
print(f"{extract_scenario_id('CR-l-0-1-0')=}")
```

```
extract_scenario_id('CR-ml-0-1-0')='ml'
extract_scenario_id('CR-l-0-1-0')='l'
```

These scenario IDs can then be used to find details of the complete scenario. These details will be provided both via the CMIP CVs (see https://github.com/WCRP-CMIP/CMIP7-CVs) and the final ScenarioMIP paper (revisions of this paper are expected soon). As above, note that the scenario IDs have changed since publication of the draft dataset. Table \ref{tab:scenario_ids} provides an overview of the changes. Scenario IDs of the final datasets can be confirmed here.

Table 4.1: Mapping of draft scenario IDs to final scenario IDs.

\begin{table}[ht] \centering \caption{Mapping of draft scenario IDs to final scenario IDs} \label{tab:scenario_ids} \begin{tabular}{cc} \hline Draft dataset & Final dataset \ \hline \textbf{vllo} & \textbf{vl} \ \textbf{vlho} & \textbf{ln} \ l & l \ ml & ml \ m & m \ hl & hl \ h & h \ \hline \end{tabular} \end{table}

1.4.3 Grids and frequencies provided

We provide five combinations of grids and time sampling (also referred to as *frequency*, although this is a bit of a misuse as the units of frequency are per time, which doesn't match the convention for these metadata values). The grid and frequency information for each file can be found in its netCDF header under the attributes `grid_label` (for grid) and `frequency` (for time sampling). The `grid_label` and `frequency` also appear in each file's name, which allows files to be filtered without needing to load them first. The five combinations of grid and time sampling are:

1. global-, annual-mean (`grid_label="gm"`, `frequency="yr"`)
2. global-, monthly-mean (`grid_label="gm"`, `frequency="mon"`)
3. hemispheric-, annual-mean (`grid_label="gr1z"`, `frequency="yr"`)
4. hemispheric-, monthly-mean (`grid_label="gr1z"`, `frequency="mon"`)
5. 15-degree latitudinal, monthly-mean (`grid_label="gnz"`, `frequency="mon"`)

1.4.4 Species provided

We provide concentrations for 43 greenhouse gas concentrations and species, as well as three equivalent species. The species are:

- major greenhouse gases (3)
  - CH4, CO2, N2O
- ozone-depleting substances (17)
  - CFCs (5)
    - CFC-11, CFC-113, CFC-114, CFC-115, CFC-12
  - HCFCs (3)
    - HCFC-141b, HCFC-142b, HCFC-22
  - Halons (3)
    - Halon 1211, Halon 1301, Halon 2402
  - other ozone-depleting substances (6)
    - CCl4, CH2Cl2, CH3Br, CH3CCl3, CH3Cl, CHCl3
- ozone fluorinated compounds (23)
  - HFCs (11)
    - HFC-125, HFC-134a, HFC-143a, HFC-152a, HFC-227ea, HFC-23, HFC-236fa, HFC-245fa, HFC-32, HFC-365mfc, HFC-4310mee
  - PFCs (9)
    - C2F6, C3F8, C4F10, C5F12, C6F14, C7F16, C8F18, CC4F8, CF4
  - other (3)
    - NF3, SF6, SO2F2

Equivalent species

For most models, you will not use all 43 species. As a result, we provide equivalent species too. There are two options if you don't want to use all 43 species.

Option 1

Use CO2, CH4, N2O and CFC-12 directly. Use CFC-11 equivalent to capture the radiative effect of all other species.

Option 2

Use CO2, CH4 and N2O directly. Use CFC-12 equivalent to capture the radiative effect of all ozone depleting substances (ODSs) and HFC-134a equivalent to capture the radiative effect of all other fluorinated gases.

1.4.5 Uncertainty

At present, we provide no analysis of the uncertainty associated with these datasets.

On top of the uncertainties in the historical data (which are, in radiative forcing terms, small), the scenario datasets are subject to uncertainties from the modelling process required to produce them. This means that, in radiative forcing terms, the uncertainty in these concentrations is much larger than the historical data. Nonetheless, it is very likely to be small compared to other uncertainties in the climate system, but this statement is not presently based on any robust analysis (rather it is based on expert judgement and past experience).

1.4.6 Differences compared to CMIP6

There are two major differences from CMIP6. The first is that the scenarios are different. By definition, this changes the concentrations. (There is also no 1:1 mapping between CMIP6 and CMIP7 scenarios, so the overall spacing and ensemble of scenarios is different too). The second is the transition from the history, observation-based period to the scenario, model-based projections. In CMIP6, this transition simply occured over a single year. In places, this led to notable changes in gradient at this transition point (further analysis of this is provided below). In CMIP7, we instead use a more sophisticated harmonisation algorithm (using the gradient-aware-harmonisation package developed as part of the ESA project). This leads to smoother, more realistic transitions between the history, observation-based period and the scenario, model-based projections.

1.5 User guide

Having downloaded the data, using it is relatively straightforward (scenario identification issue discussed above notwithstanding).

1.5.1 Annual-, global-mean data

We start with the annual-, global-mean data. Like all our datasets, this is composed of one file per scenario, given seven files in total.

For yearly data, the time labels in the filename are years (for months, the month is included e.g. you will see `202201-210012` rather than `2022-2100` in the filename, the files also have different values for the `frequency` attribute). Global-mean data is identified by the 'grid label' `gm`, which appears in the filename. Below we show the filenames for the CO2 output.

Note: in the draft datasets, the time axis starts in 2023. This will be updated to a 2022 start for the final datasets, in line with the rest of the scenario datasets.

```
- co2_input4MIPs_GHGConcentrations_ScenarioMIP_CR-vllo-0-1-0_gm_2023-2100.nc
- co2_input4MIPs_GHGConcentrations_ScenarioMIP_CR-vlho-0-1-0_gm_2023-2100.nc
- co2_input4MIPs_GHGConcentrations_ScenarioMIP_CR-ml-0-1-0_gm_2023-2100.nc
- co2_input4MIPs_GHGConcentrations_ScenarioMIP_CR-m-0-1-0_gm_2023-2100.nc
- co2_input4MIPs_GHGConcentrations_ScenarioMIP_CR-l-0-1-0_gm_2023-2100.nc
- co2_input4MIPs_GHGConcentrations_ScenarioMIP_CR-hl-0-1-0_gm_2023-2100.nc
- co2_input4MIPs_GHGConcentrations_ScenarioMIP_CR-h-0-1-0_gm_2023-2100.nc
```

Output for other gases are named identically, with `co2` being replaced by the other gas name. For example, for methane the filenames are:

```
- ch4_input4MIPs_GHGConcentrations_ScenarioMIP_CR-vllo-0-1-0_gm_2023-2100.nc
- ch4_input4MIPs_GHGConcentrations_ScenarioMIP_CR-vlho-0-1-0_gm_2023-2100.nc
- ch4_input4MIPs_GHGConcentrations_ScenarioMIP_CR-ml-0-1-0_gm_2023-2100.nc
- ch4_input4MIPs_GHGConcentrations_ScenarioMIP_CR-m-0-1-0_gm_2023-2100.nc
- ch4_input4MIPs_GHGConcentrations_ScenarioMIP_CR-l-0-1-0_gm_2023-2100.nc
- ch4_input4MIPs_GHGConcentrations_ScenarioMIP_CR-hl-0-1-0_gm_2023-2100.nc
- ch4_input4MIPs_GHGConcentrations_ScenarioMIP_CR-h-0-1-0_gm_2023-2100.nc
```

As described above, the data is netCDF files. This means that metadata can be trivially inspected using a tool like `ncdump`. There is a lot of metadata included in these files. In general, you should not need to parse this metadata directly. However, if you have specific questions, please feel free to contact the emails given in the `contact` attribute.

```
Der Befehl "fold" ist entweder falsch geschrieben oder
konnte nicht gefunden werden.
```

Using a tool like xarray, loading and working with the data is trivial.

```
import xarray as xr

time_coder = xr.coders.CFDatetimeCoder(use_cftime=True)
ds_example_co2_yearly_global = xr.open_dataset(
    co2_yearly_global_fps[-1], decode_times=time_coder
)
```

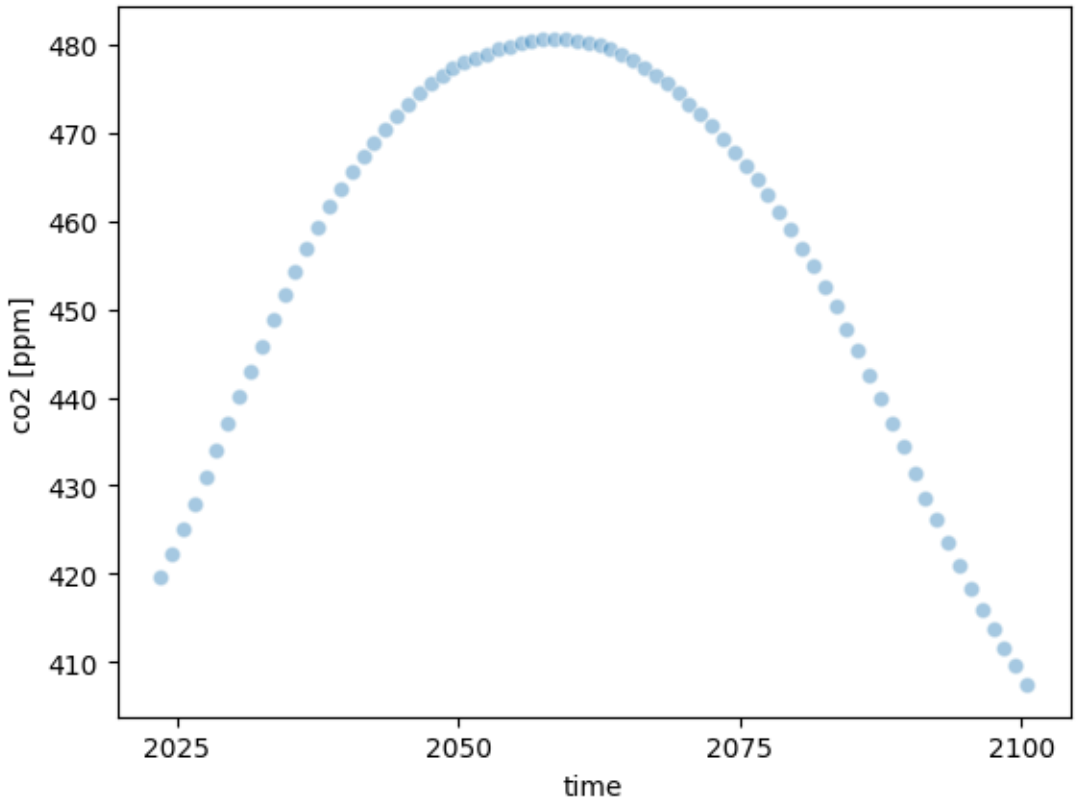
```
ds_example_co2_yearly_global
```

```
<xarray.Dataset> Size: 2kB
Dimensions:    (time: 78, bnds: 2)
Coordinates:
  * time       (time) object 624B 2023-07-02 12:00:00 ... 2100-07-02 12:00:00
Dimensions without coordinates: bnds
Data variables:
    co2        (time) float32 312B 419.6 422.2 425.0 428.0 ... 411.7 409.6 407.5
```

(continues on next page)

(continued from previous page)

```
    time_bnds  (time, bnds) object 1kB 2023-01-01 00:00:00 ... 2101-01-01 00:...
Attributes: (12/29)
    Conventions:             CF-1.7
    activity_id:             input4MIPs
    comment:                 Data compiled by Climate Resource, based on scie...
    contact:                 zebedee.nicholls@climate-resource.com;florence.b...
    creation_date:           2025-09-18T10:30:26Z
    dataset_category:        GHGConcentrations
    ...                      ...
    references_urls:         https://github.com/climate-resource/CMIP-GHG-Con...
    source_id:               CR-vlho-0-1-0
    source_version:          0.1.0
    target_mip:              ScenarioMIP
    tracking_id:             hdl:21.14100/07393674-0b97-4f82-b503-24e01b4dca7d
    variable_id:             co2
```



1.5.2 Space- and time-average nature of the data

All of our data represents the mean over each cell. This is indicated by the `cell_methods` attribute of all of our output variables.

```
ds_example_co2_yearly_global["co2"].attrs["cell_methods"]
```

```
'area: time: mean'
```

This mean is both in space and time. The time bounds covered by each step are specified by the `time_bnds` variable

(when there is spatial information, equivalent `lat_bnds` and `lon_bnds` information is also included). This variable specifies the start (inclusive) and end (exclusive) of the time period covered by each data point.

```
ds_example_co2_yearly_global["time_bnds"]
```

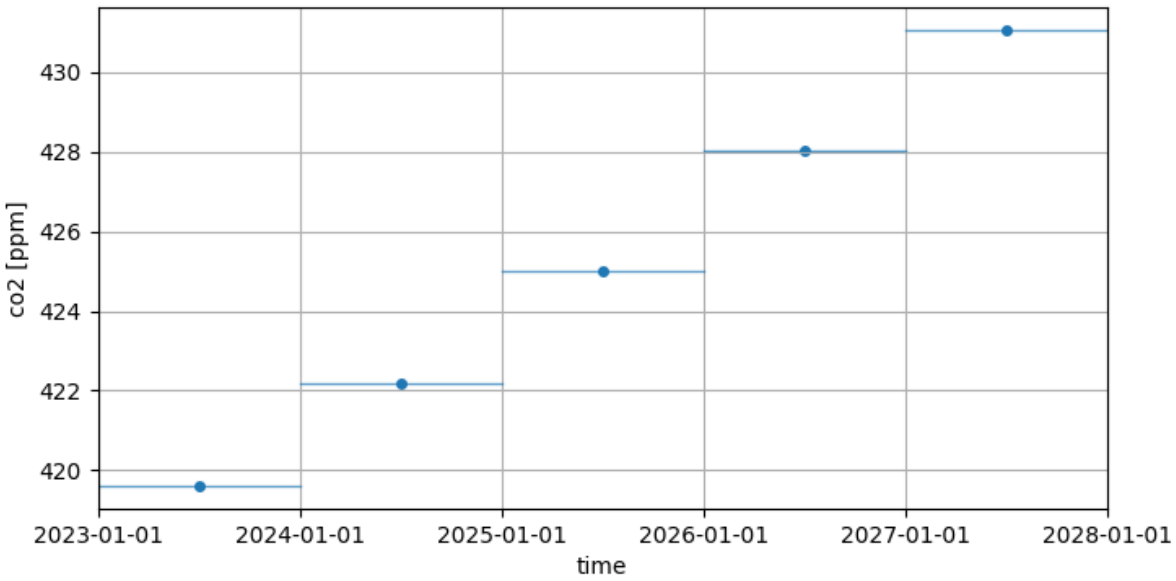
```
<xarray.DataArray 'time_bnds' (time: 78, bnds: 2)> Size: 1kB
array([[cftime.DatetimeProlepticGregorian(2023, 1, 1, 0, 0, 0, 0, has_year_
 ↪zero=True),
        cftime.DatetimeProlepticGregorian(2024, 1, 1, 0, 0, 0, 0, has_year_
 ↪zero=True)],
       [cftime.DatetimeProlepticGregorian(2024, 1, 1, 0, 0, 0, 0, has_year_
 ↪zero=True),
        cftime.DatetimeProlepticGregorian(2025, 1, 1, 0, 0, 0, 0, has_year_
 ↪zero=True)],
       [cftime.DatetimeProlepticGregorian(2025, 1, 1, 0, 0, 0, 0, has_year_
 ↪zero=True),
        cftime.DatetimeProlepticGregorian(2026, 1, 1, 0, 0, 0, 0, has_year_
 ↪zero=True)],
       [cftime.DatetimeProlepticGregorian(2026, 1, 1, 0, 0, 0, 0, has_year_
 ↪zero=True),
        cftime.DatetimeProlepticGregorian(2027, 1, 1, 0, 0, 0, 0, has_year_
 ↪zero=True)],
       [cftime.DatetimeProlepticGregorian(2027, 1, 1, 0, 0, 0, 0, has_year_
 ↪zero=True),
        cftime.DatetimeProlepticGregorian(2028, 1, 1, 0, 0, 0, 0, has_year_
 ↪zero=True)],
       [cftime.DatetimeProlepticGregorian(2028, 1, 1, 0, 0, 0, 0, has_year_
 ↪zero=True),
        cftime.DatetimeProlepticGregorian(2029, 1, 1, 0, 0, 0, 0, has_year_
 ↪zero=True)],
       [cftime.DatetimeProlepticGregorian(2029, 1, 1, 0, 0, 0, 0, has_year_
 ↪zero=True),
        cftime.DatetimeProlepticGregorian(2030, 1, 1, 0, 0, 0, 0, has_year_
 ↪zero=True)],
       [cftime.DatetimeProlepticGregorian(2030, 1, 1, 0, 0, 0, 0, has_year_
 ↪zero=True),
        cftime.DatetimeProlepticGregorian(2031, 1, 1, 0, 0, 0, 0, has_year_
 ↪zero=True)],
       [cftime.DatetimeProlepticGregorian(2031, 1, 1, 0, 0, 0, 0, has_year_
 ↪zero=True),
        cftime.DatetimeProlepticGregorian(2032, 1, 1, 0, 0, 0, 0, has_year_
 ↪zero=True)],
       [cftime.DatetimeProlepticGregorian(2032, 1, 1, 0, 0, 0, 0, has_year_
 ↪zero=True),
        cftime.DatetimeProlepticGregorian(2033, 1, 1, 0, 0, 0, 0, has_year_
 ↪zero=True)],
...
        cftime.DatetimeProlepticGregorian(2092, 1, 1, 0, 0, 0, 0, has_year_
 ↪zero=True)],
       [cftime.DatetimeProlepticGregorian(2092, 1, 1, 0, 0, 0, 0, has_year_
 ↪zero=True),
        cftime.DatetimeProlepticGregorian(2093, 1, 1, 0, 0, 0, 0, has_year_
 ↪zero=True)],
       [cftime.DatetimeProlepticGregorian(2093, 1, 1, 0, 0, 0, 0, has_year_
 ↪zero=True),
        cftime.DatetimeProlepticGregorian(2094, 1, 1, 0, 0, 0, 0, has_year_
 ↪zero=True)],
       [cftime.DatetimeProlepticGregorian(2094, 1, 1, 0, 0, 0, 0, has_year_
```

(continues on next page)

(continued from previous page)

```
↪zero=True),
       cftime.DatetimeProlepticGregorian(2095, 1, 1, 0, 0, 0, 0, has_year_
↪zero=True)],
      [cftime.DatetimeProlepticGregorian(2095, 1, 1, 0, 0, 0, 0, has_year_
↪zero=True),
       cftime.DatetimeProlepticGregorian(2096, 1, 1, 0, 0, 0, 0, has_year_
↪zero=True)],
      [cftime.DatetimeProlepticGregorian(2096, 1, 1, 0, 0, 0, 0, has_year_
↪zero=True),
       cftime.DatetimeProlepticGregorian(2097, 1, 1, 0, 0, 0, 0, has_year_
↪zero=True)],
      [cftime.DatetimeProlepticGregorian(2097, 1, 1, 0, 0, 0, 0, has_year_
↪zero=True),
       cftime.DatetimeProlepticGregorian(2098, 1, 1, 0, 0, 0, 0, has_year_
↪zero=True)],
      [cftime.DatetimeProlepticGregorian(2098, 1, 1, 0, 0, 0, 0, has_year_
↪zero=True),
       cftime.DatetimeProlepticGregorian(2099, 1, 1, 0, 0, 0, 0, has_year_
↪zero=True)],
      [cftime.DatetimeProlepticGregorian(2099, 1, 1, 0, 0, 0, 0, has_year_
↪zero=True),
       cftime.DatetimeProlepticGregorian(2100, 1, 1, 0, 0, 0, 0, has_year_
↪zero=True)],
      [cftime.DatetimeProlepticGregorian(2100, 1, 1, 0, 0, 0, 0, has_year_
↪zero=True),
       cftime.DatetimeProlepticGregorian(2101, 1, 1, 0, 0, 0, 0, has_year_
↪zero=True)]],
     dtype=object)
Coordinates:
  * time     (time) object 624B 2023-07-02 12:00:00 ... 2100-07-02 12:00:00
Dimensions without coordinates: bnds
```

As a result of the time average that the data represents, it is inappropriate to plot this data using a line plot (the mean of the lines joining the points is not the same as the data given in the files). Instead, the data should be plotted (and used) as a scatter or a step plot, as shown below. (The same logic applies to any spatial plots which could be created from our datasets that include spatial dimensions).



1.5.3 Monthly-, global-mean data

If you want to have information at a finer level of temporal detail, we also provide monthly files.

For monthly data, the time labels in the filename are months. Below we show the filenames for the CO2 output.

```
- co2_input4MIPs_GHGConcentrations_ScenarioMIP_CR-vllo-0-1-0_gm_202301-210012.nc
- co2_input4MIPs_GHGConcentrations_ScenarioMIP_CR-vlho-0-1-0_gm_202301-210012.nc
- co2_input4MIPs_GHGConcentrations_ScenarioMIP_CR-ml-0-1-0_gm_202301-210012.nc
- co2_input4MIPs_GHGConcentrations_ScenarioMIP_CR-m-0-1-0_gm_202301-210012.nc
- co2_input4MIPs_GHGConcentrations_ScenarioMIP_CR-l-0-1-0_gm_202301-210012.nc
- co2_input4MIPs_GHGConcentrations_ScenarioMIP_CR-hl-0-1-0_gm_202301-210012.nc
- co2_input4MIPs_GHGConcentrations_ScenarioMIP_CR-h-0-1-0_gm_202301-210012.nc
```

Again, the data can be trivially loaded with xarray.

```
ds_example_co2_monthly_global = xr.open_mfdataset(
    co2_monthly_global_fps[-1], decode_times=time_coder
)
```

```
ds_example_co2_monthly_global
```

```
<xarray.Dataset> Size: 26kB
Dimensions:    (time: 936, bnds: 2)
Coordinates:
  * time       (time) object 7kB 2023-01-15 00:00:00 ... 2100-12-15 00:00:00
Dimensions without coordinates: bnds
Data variables:
    co2        (time) float32 4kB 418.8 418.9 419.0 419.2 ... 397.9 397.8 397.7
    time_bnds  (time, bnds) object 15kB 2023-01-01 00:00:00 ... 2101-01-01 00...
Attributes: (12/29)
    Conventions:             CF-1.7
    activity_id:             input4MIPs
    comment:                 Data compiled by Climate Resource, based on scie...
    contact:                 zebedee.nicholls@climate-resource.com;florence.b...
    creation_date:           2025-09-18T10:29:12Z
    dataset_category:        GHGConcentrations
    ...                      ...
    references_urls:         https://github.com/climate-resource/CMIP-GHG-Con...
    source_id:               CR-vllo-0-1-0
    source_version:          0.1.0
    target_mip:              ScenarioMIP
    tracking_id:             hdl:21.14100/62970cd4-d879-4b3d-a794-d8f55cfb7b28
    variable_id:             co2
```

For this data, the time bounds show that each point is the average a month, not a year.

```
ds_example_co2_monthly_global["time_bnds"]
```

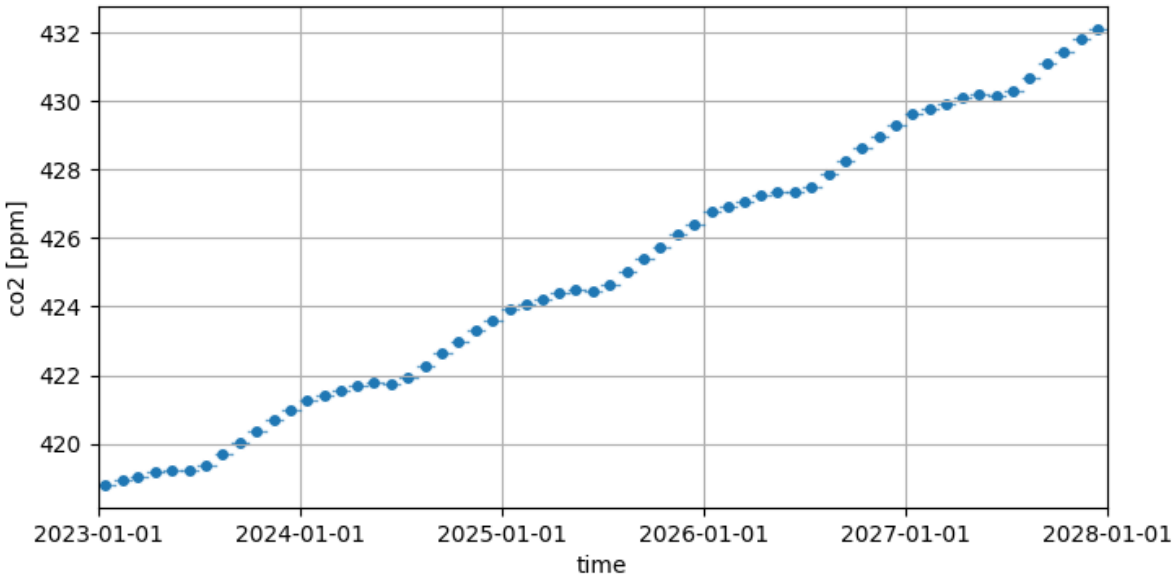
```
<xarray.DataArray 'time_bnds' (time: 936, bnds: 2)> Size: 15kB
array([[cftime.DatetimeProlepticGregorian(2023, 1, 1, 0, 0, 0, 0, has_year_
 ↪zero=True),
        cftime.DatetimeProlepticGregorian(2023, 2, 1, 0, 0, 0, 0, has_year_
 ↪zero=True)],
       [cftime.DatetimeProlepticGregorian(2023, 2, 1, 0, 0, 0, 0, has_year_
 ↪zero=True),
        cftime.DatetimeProlepticGregorian(2023, 3, 1, 0, 0, 0, 0, has_year_
```

(continues on next page)

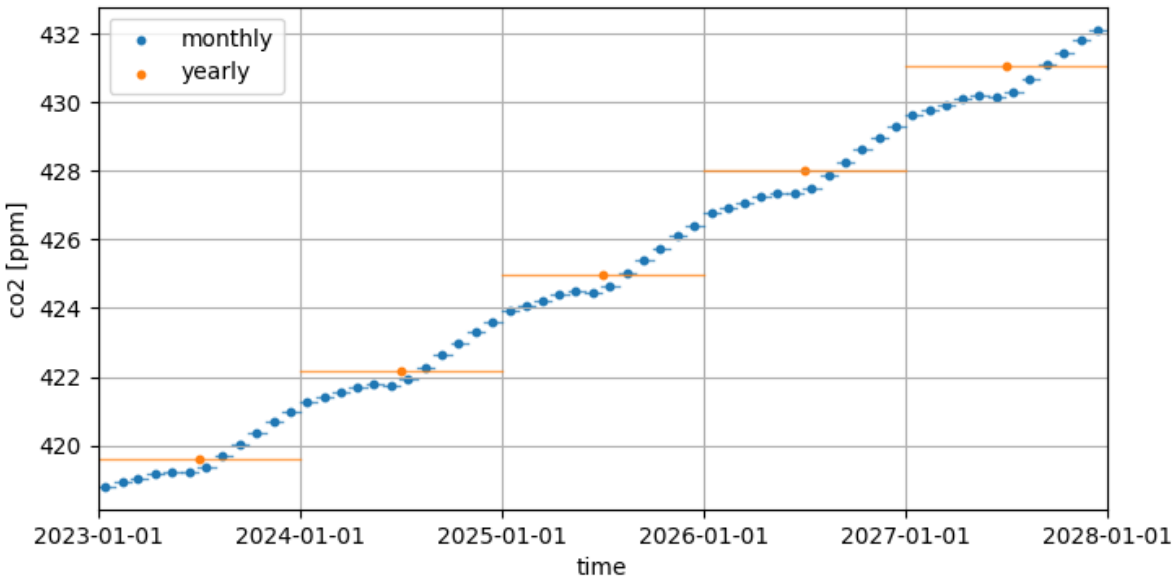
(continued from previous page)

```
↪zero=True)],
       [cftime.DatetimeProlepticGregorian(2023, 3, 1, 0, 0, 0, 0, has_year_
↪zero=True),
        cftime.DatetimeProlepticGregorian(2023, 4, 1, 0, 0, 0, 0, has_year_
↪zero=True)],
       ...,
       [cftime.DatetimeProlepticGregorian(2100, 10, 1, 0, 0, 0, 0, has_year_
↪zero=True),
        cftime.DatetimeProlepticGregorian(2100, 11, 1, 0, 0, 0, 0, has_year_
↪zero=True)],
       [cftime.DatetimeProlepticGregorian(2100, 11, 1, 0, 0, 0, 0, has_year_
↪zero=True),
        cftime.DatetimeProlepticGregorian(2100, 12, 1, 0, 0, 0, 0, has_year_
↪zero=True)],
       [cftime.DatetimeProlepticGregorian(2100, 12, 1, 0, 0, 0, 0, has_year_
↪zero=True),
        cftime.DatetimeProlepticGregorian(2101, 1, 1, 0, 0, 0, 0, has_year_
↪zero=True)]],
      shape=(936, 2), dtype=object)
Coordinates:
  * time     (time) object 7kB 2023-01-15 00:00:00 ... 2100-12-15 00:00:00
Dimensions without coordinates: bnds
```

As above, as a result of the time average that the data represents, it is inappropriate to plot this data using a line plot. Scatter or step plots should be used instead.



The monthly data includes seasonality. Plotting the monthly and yearly data on the same axes makes particularly clear why a line plot is inappropriate.



1.5.4 Monthly-, latitudinally-resolved data

We also provide data with spatial, specifically latitudinal, resolution. This data comes on a 15-degree latitudinal grid (see below for details of the grid and latitudinal bounds). These files are identified by the grid label `gnz`. We only provide these files with monthly resolution.

For completeness, we note that we also provide hemispheric means. These are not shown here, but are identified by the grid label `gr1z`.

Below we show the filenames for the latitudinally-resolved data for CO2

```
- co2_input4MIPs_GHGConcentrations_ScenarioMIP_CR-vllo-0-1-0_gnz_202301-210012.nc
- co2_input4MIPs_GHGConcentrations_ScenarioMIP_CR-vlho-0-1-0_gnz_202301-210012.nc
- co2_input4MIPs_GHGConcentrations_ScenarioMIP_CR-ml-0-1-0_gnz_202301-210012.nc
- co2_input4MIPs_GHGConcentrations_ScenarioMIP_CR-m-0-1-0_gnz_202301-210012.nc
- co2_input4MIPs_GHGConcentrations_ScenarioMIP_CR-l-0-1-0_gnz_202301-210012.nc
- co2_input4MIPs_GHGConcentrations_ScenarioMIP_CR-hl-0-1-0_gnz_202301-210012.nc
- co2_input4MIPs_GHGConcentrations_ScenarioMIP_CR-h-0-1-0_gnz_202301-210012.nc
```

Again, the data can be trivially loaded with xarray.

```
ds_example_co2_monthly_lat = xr.open_mfdataset(
    co2_monthly_lat_fps[-1],
    decode_times=time_coder,
    data_vars=None,
    compat="no_conflicts",
)
```

```
ds_example_co2_monthly_lat
```

```
<xarray.Dataset> Size: 68kB
Dimensions:    (time: 936, lat: 12, bnds: 2)
Coordinates:
  * time       (time) object 7kB 2023-01-15 00:00:00 ... 2100-12-15 00:00:00
  * lat        (lat) float64 96B -82.5 -67.5 -52.5 -37.5 ... 37.5 52.5 67.5 82.5
```

(continues on next page)

(continued from previous page)

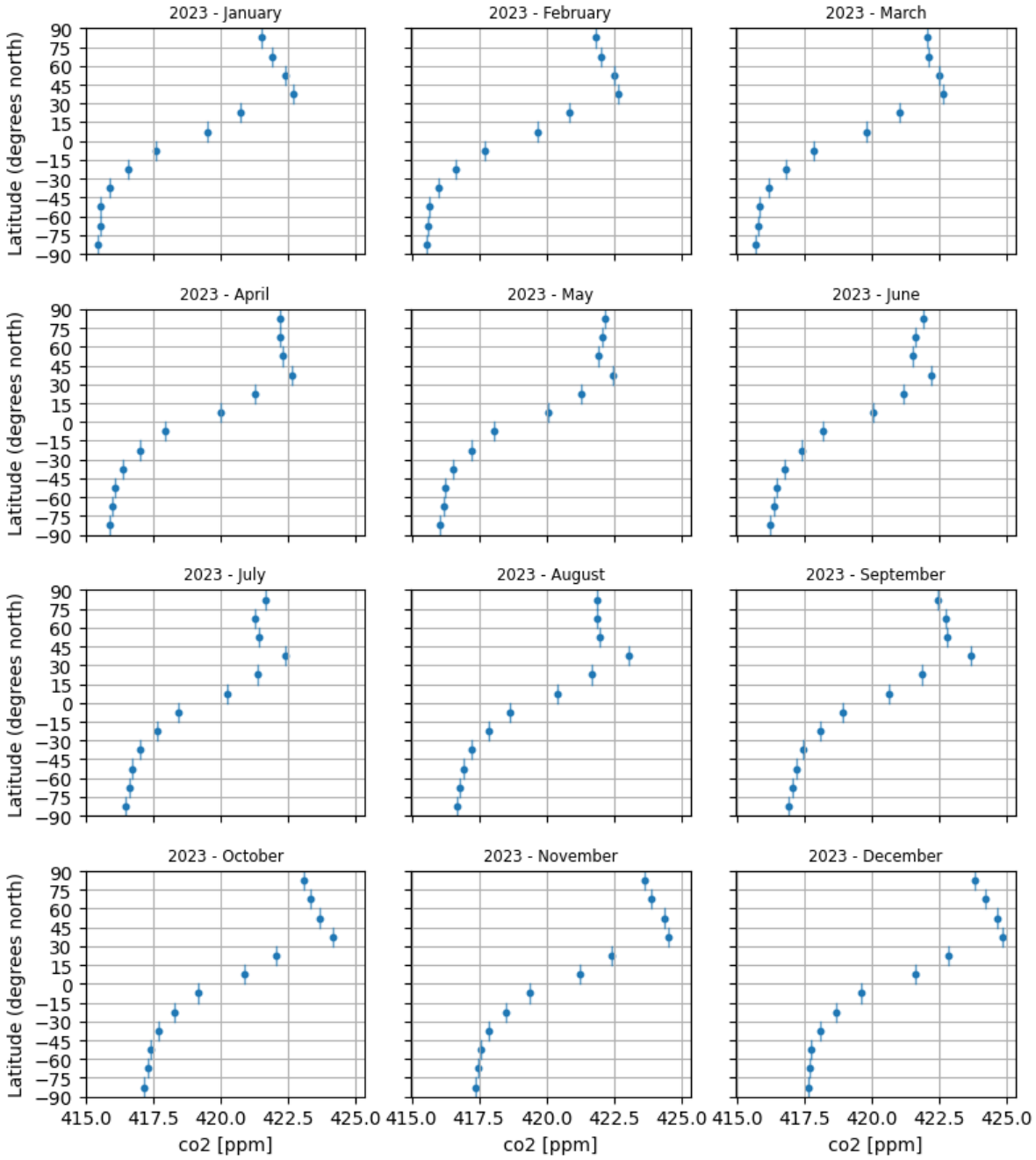
```
Dimensions without coordinates: bnds
Data variables:
    co2        (time, lat) float32 45kB 415.4 415.5 415.5 ... 704.5 702.8 700.7
    time_bnds  (time, bnds) object 15kB 2023-01-01 00:00:00 ... 2101-01-01 00...
    lat_bnds   (lat, bnds) float64 192B -90.0 -75.0 -75.0 ... 75.0 75.0 90.0
Attributes: (12/29)
    Conventions:             CF-1.7
    activity_id:             input4MIPs
    comment:                 Data compiled by Climate Resource, based on scie...
    contact:                 zebedee.nicholls@climate-resource.com;florence.b...
    creation_date:           2025-09-18T10:29:54Z
    dataset_category:        GHGConcentrations
    ...                      ...
    references_urls:         https://github.com/climate-resource/CMIP-GHG-Con...
    source_id:               CR-h-0-1-0
    source_version:          0.1.0
    target_mip:              ScenarioMIP
    tracking_id:             hdl:21.14100/febf1b5e-00a6-4f9d-9db0-4891cdd8abad
    variable_id:             co2
```

For this data, the latitudinal bounds show the area over which each point is the average.

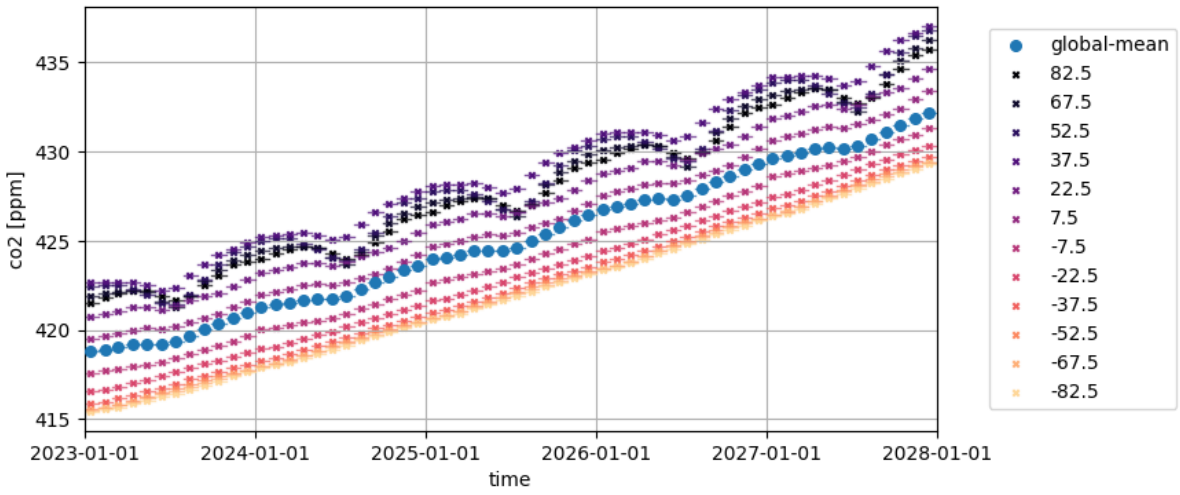
```
ds_example_co2_monthly_lat["lat_bnds"]
```

```
<xarray.DataArray 'lat_bnds' (lat: 12, bnds: 2)> Size: 192B
array([[-90., -75.],
       [-75., -60.],
       [-60., -45.],
       [-45., -30.],
       [-30., -15.],
       [-15.,   0.],
       [  0.,  15.],
       [ 15.,  30.],
       [ 30.,  45.],
       [ 45.,  60.],
       [ 60.,  75.],
       [ 75.,  90.]])
Coordinates:
  * lat      (lat) float64 96B -82.5 -67.5 -52.5 -37.5 ... 37.5 52.5 67.5 82.5
Dimensions without coordinates: bnds
```

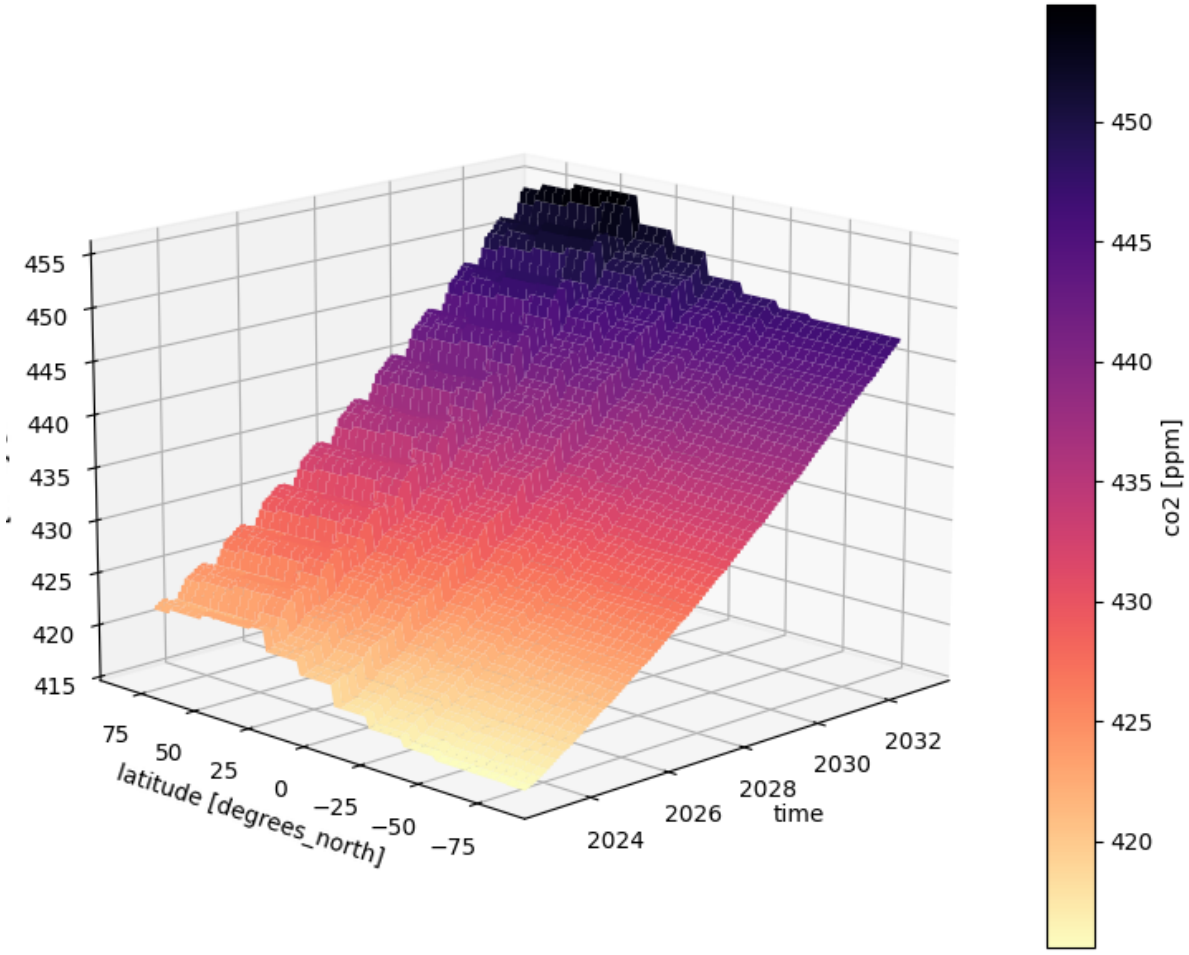
As above, but this time for the spatial axis, it is inappropriate to plot this data using a line plot. Scatter or step plots should be used instead.



We can compare the global-mean data to the data at each latitude. The strength of the latitudinal gradient varies also by gas (not shown).



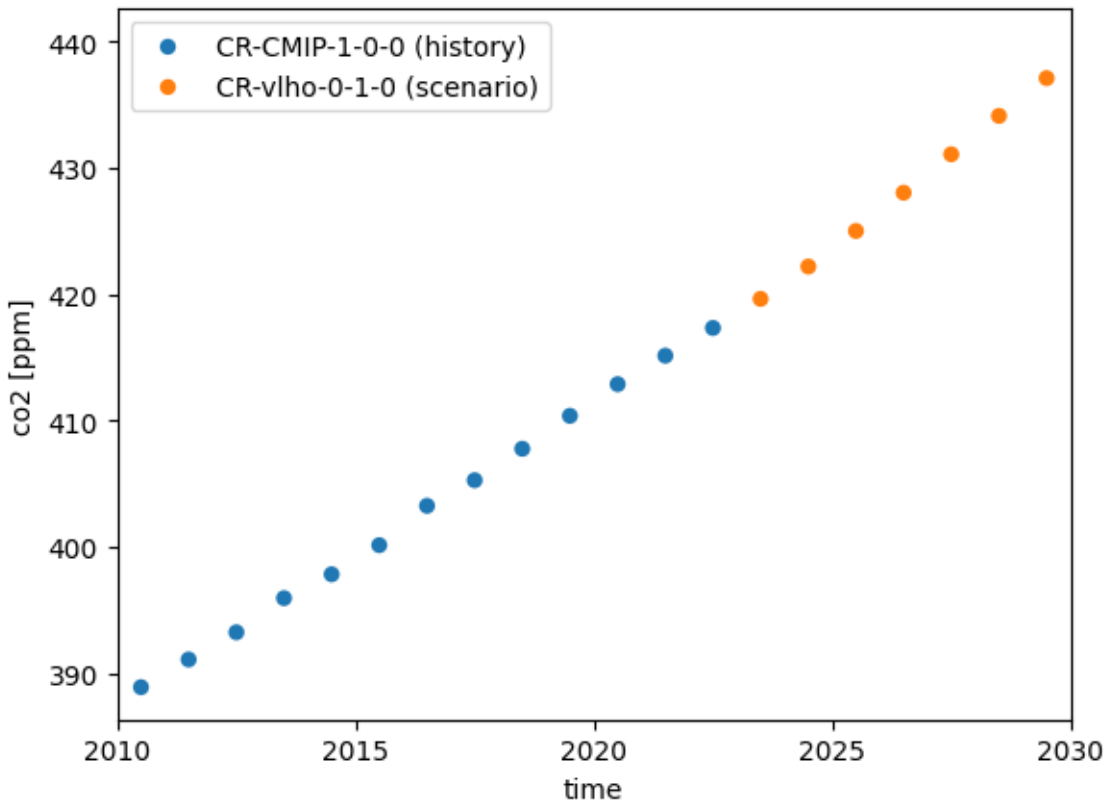
The data can also be plotted in a so-called “magic carpet” to see the variation in space and time simultaneously.



1.5.5 Transition from history

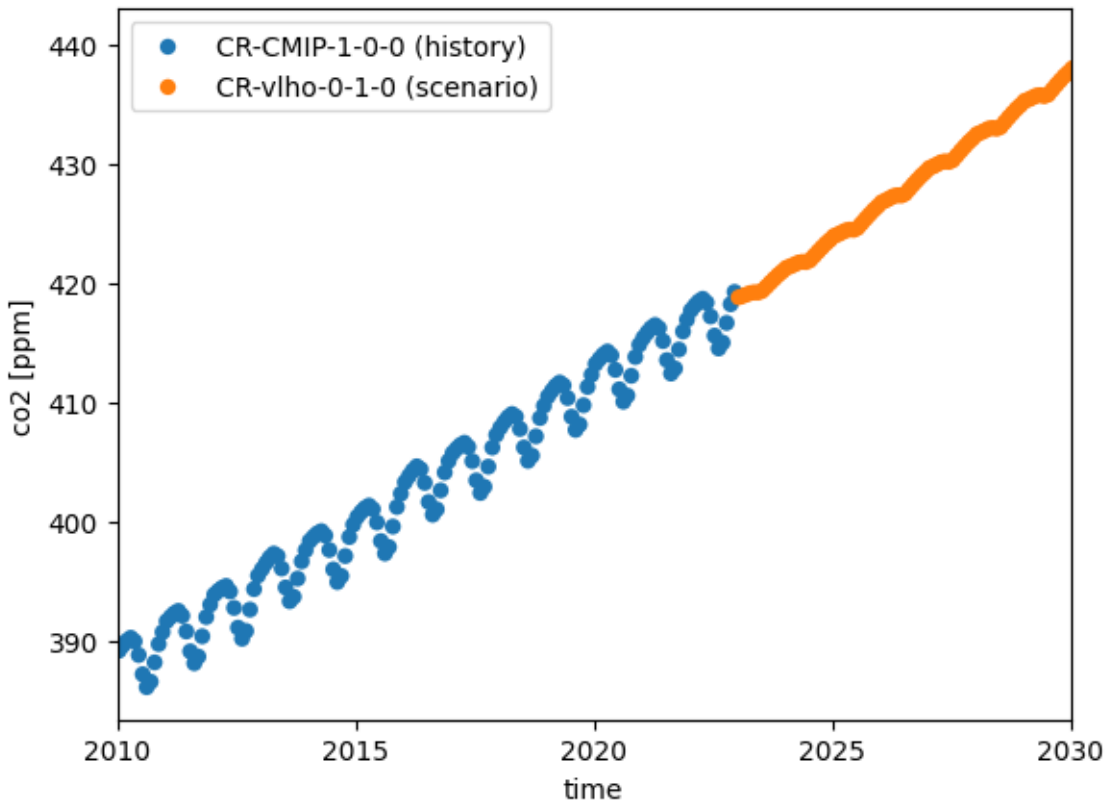
Each dataset transitions smoothly from the historical data.

1.5.6 Annual-, global-mean data



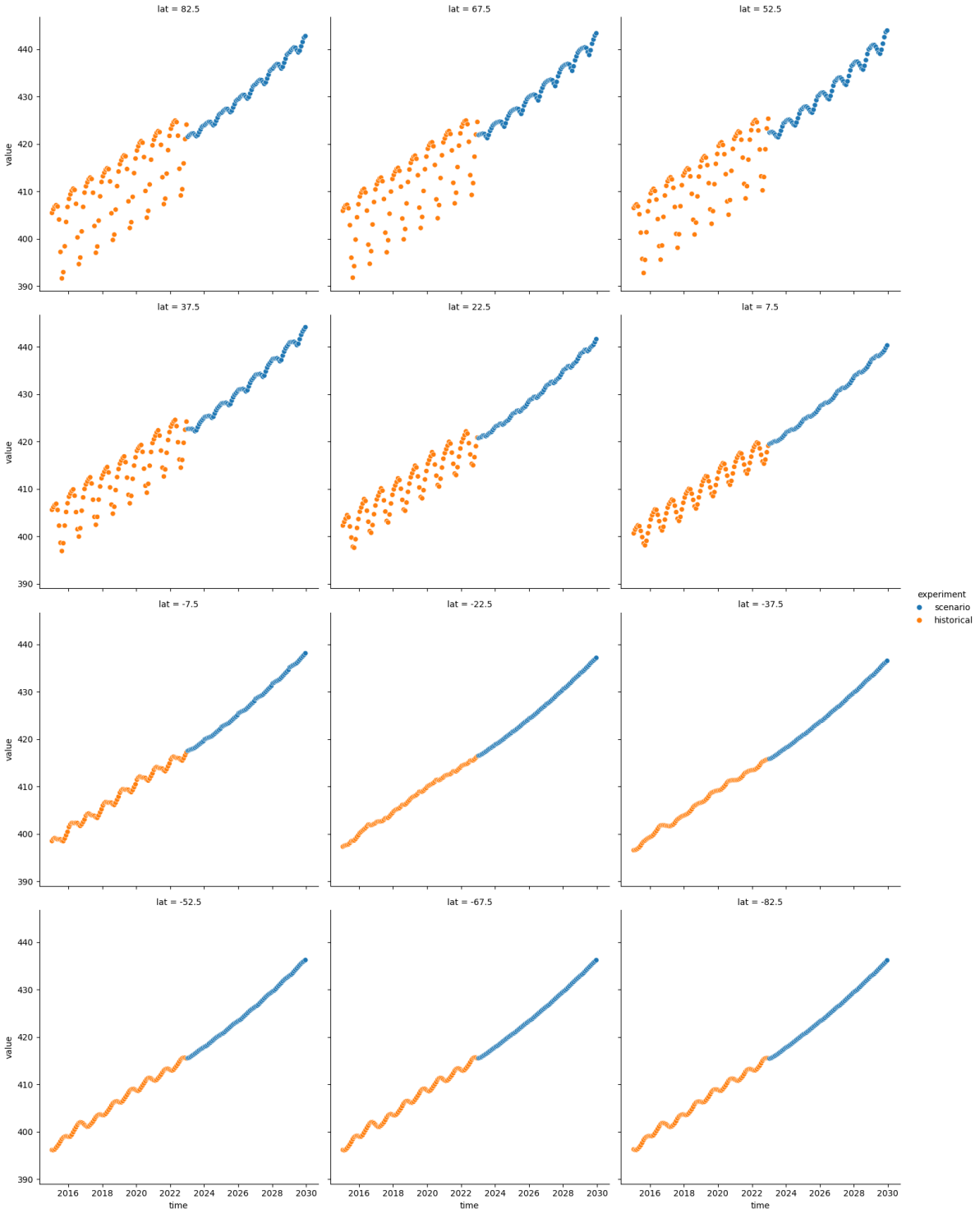
Monthly-, global-mean data

Note that the transition from history to scenarios is clearly wrong in the draft dataset. This will be fixed before the final dataset is published.



Monthly-, 15-degree latitudinally binned data

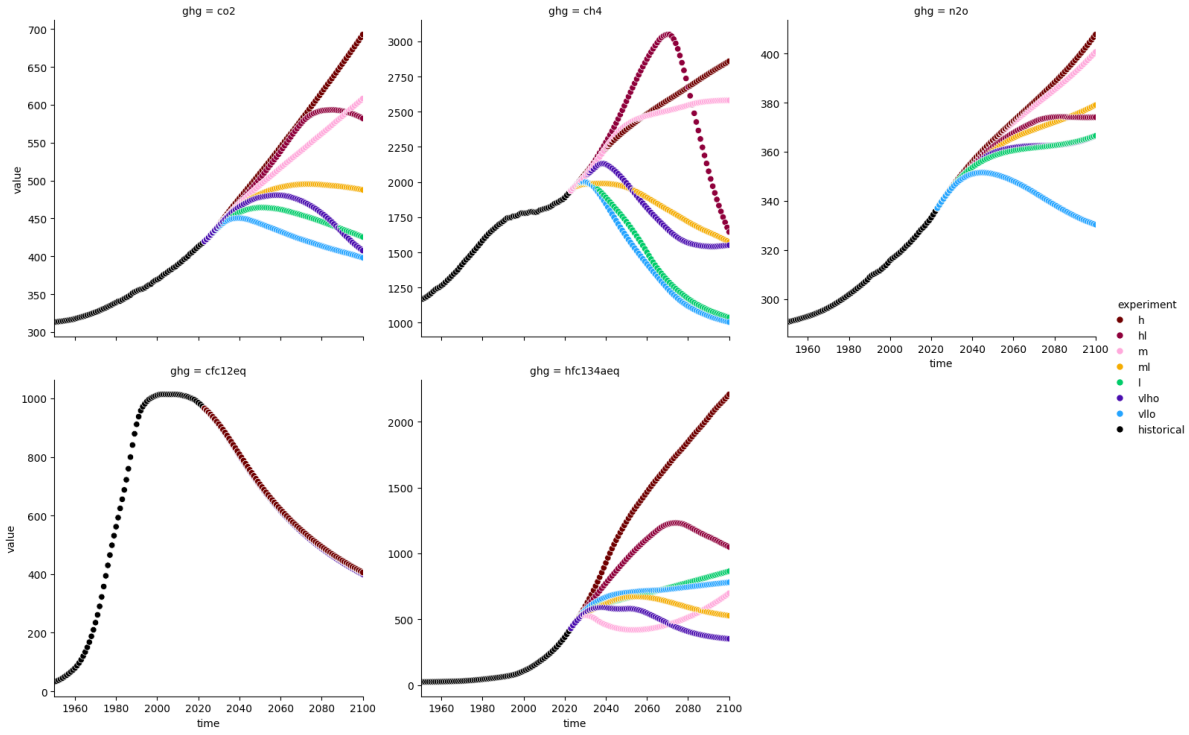
Note that the transition from history to scenarios is clearly wrong in the draft dataset. This will be fixed before the final dataset is published.


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
value
time
experiment
scenario
historical

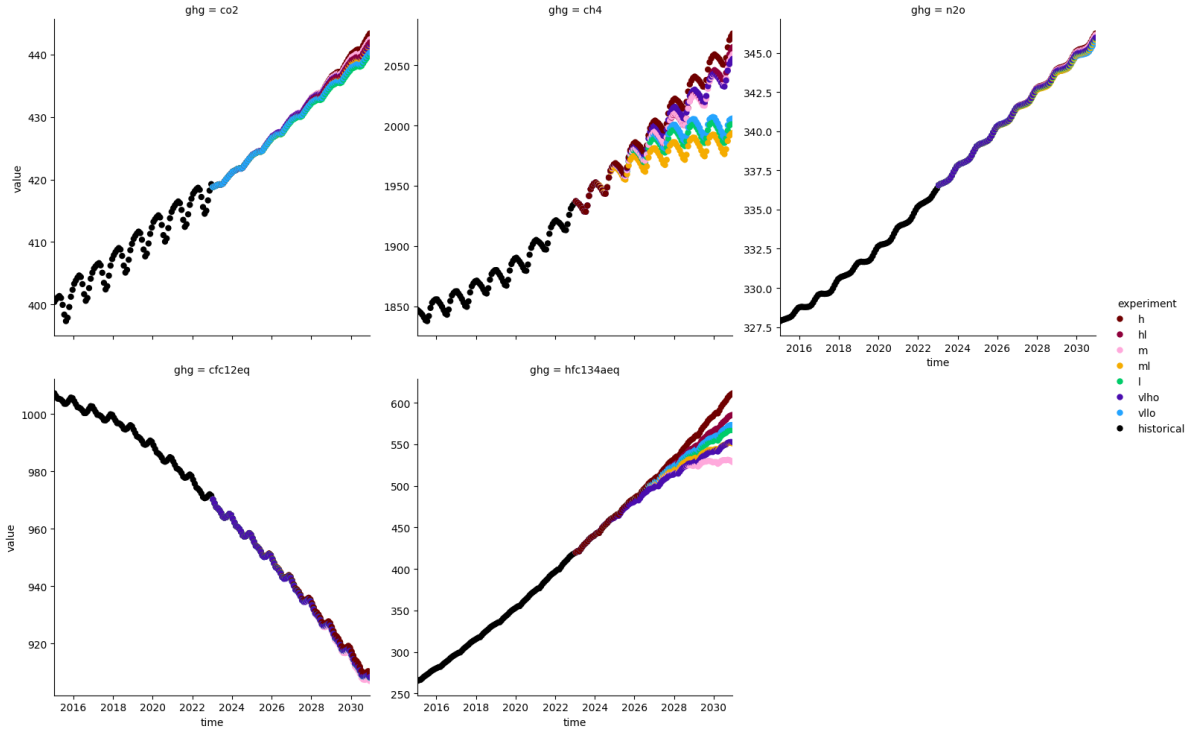
1.5.7 Full scenario set

Having seen the transition for a single scenario, we now show the full scenario set including the transition from history.

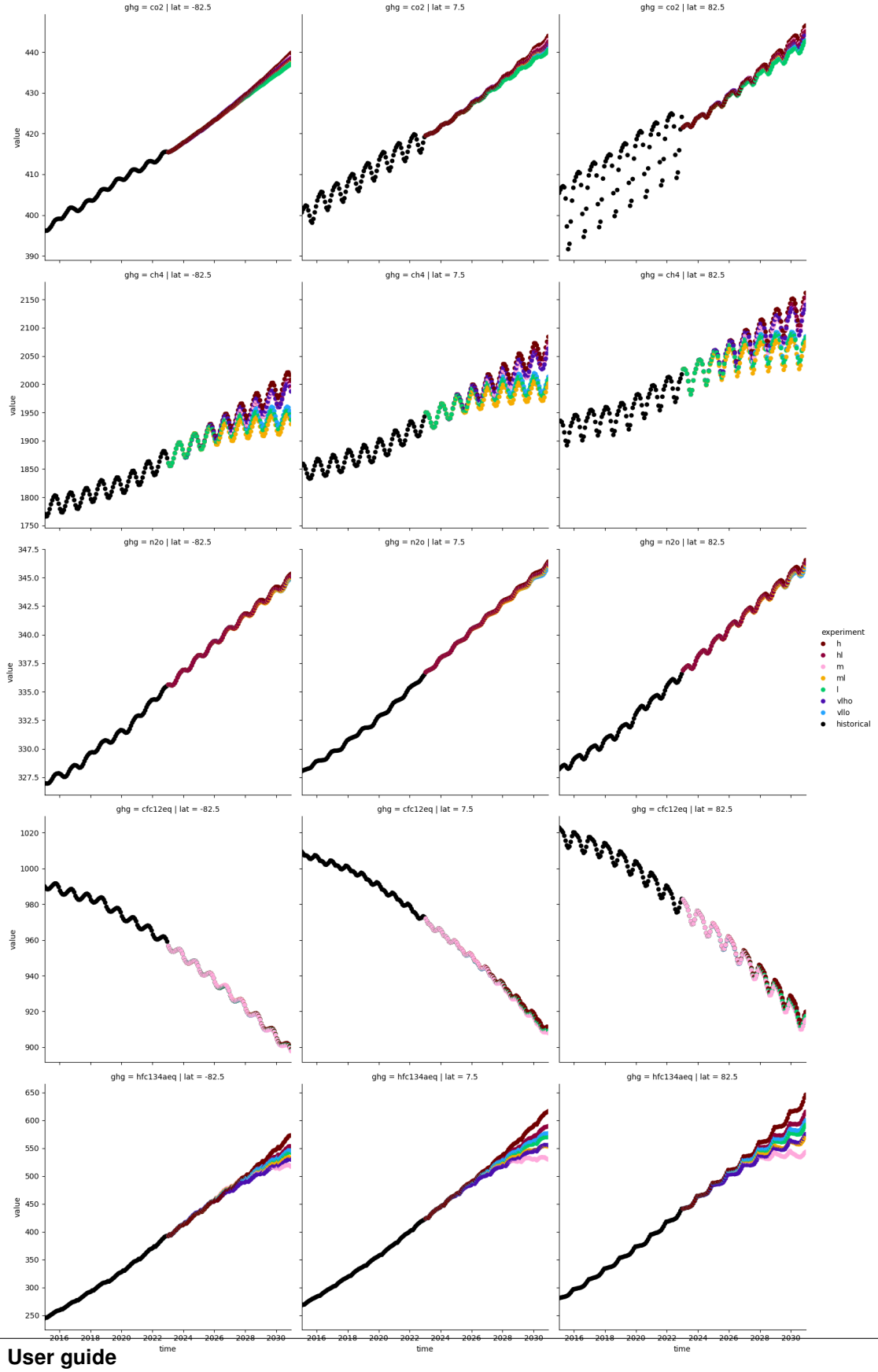
Annual-, global-mean data



Monthly-, global-mean data



Monthly-, 15-degree latitudinal data



1.5.8 Differences from CMIP6

File formats and naming

The file formats are generally close to CMIP6. There are three key changes:

1. the global-mean and hemispheric-mean data are split into separate files. In CMIP6, this data was in the same file (with a grid label of `GMNHSH`). We have split this for two reasons: a) `GMNHSH` is not a grid label recognised in the CMIP CVs and b) having global-mean and hemispheric-mean data in the same file required us to introduce a 'sector' coordinate, which was confusing and does not follow the CF-conventions.
2. the files are split into different time components. The CMIP6 data had the scenarios and their extensions in a single file. The CMIP7 extensions are not defined yet, so the scenarios (up to 2100) will be in one file, with the extensions being in a separate file (and under separate source IDs).
3. the variable names were simplified. They are now simply the names of the gases, for example we now use "co2" rather than "mole_fraction_of_carbon_dioxide". A full mapping is provided below.

Variable name mapping

```
CMIP6_TO_CMIP7_VARIABLE_MAP = {
    # name in CMIP6: name in CMIP7
    "mole_fraction_of_carbon_dioxide_in_air": "co2",
    "mole_fraction_of_methane_in_air": "ch4",
    "mole_fraction_of_nitrous_oxide_in_air": "n2o",
    "mole_fraction_of_c2f6_in_air": "c2f6",
    "mole_fraction_of_c3f8_in_air": "c3f8",
    "mole_fraction_of_c4f10_in_air": "c4f10",
    "mole_fraction_of_c5f12_in_air": "c5f12",
    "mole_fraction_of_c6f14_in_air": "c6f14",
    "mole_fraction_of_c7f16_in_air": "c7f16",
    "mole_fraction_of_c8f18_in_air": "c8f18",
    "mole_fraction_of_c_c4f8_in_air": "cc4f8",
    "mole_fraction_of_carbon_tetrachloride_in_air": "ccl4",
    "mole_fraction_of_cf4_in_air": "cf4",
    "mole_fraction_of_cfc11_in_air": "cfc11",
    "mole_fraction_of_cfc113_in_air": "cfc113",
    "mole_fraction_of_cfc114_in_air": "cfc114",
    "mole_fraction_of_cfc115_in_air": "cfc115",
    "mole_fraction_of_cfc12_in_air": "cfc12",
    "mole_fraction_of_ch2cl2_in_air": "ch2cl2",
    "mole_fraction_of_methyl_bromide_in_air": "ch3br",
    "mole_fraction_of_ch3ccl3_in_air": "ch3ccl3",
    "mole_fraction_of_methyl_chloride_in_air": "ch3cl",
    "mole_fraction_of_chcl3_in_air": "chcl3",
    "mole_fraction_of_halon1211_in_air": "halon1211",
    "mole_fraction_of_halon1301_in_air": "halon1301",
    "mole_fraction_of_halon2402_in_air": "halon2402",
    "mole_fraction_of_hcfc141b_in_air": "hcfc141b",
    "mole_fraction_of_hcfc142b_in_air": "hcfc142b",
    "mole_fraction_of_hcfc22_in_air": "hcfc22",
    "mole_fraction_of_hfc125_in_air": "hfc125",
    "mole_fraction_of_hfc134a_in_air": "hfc134a",
    "mole_fraction_of_hfc143a_in_air": "hfc143a",
    "mole_fraction_of_hfc152a_in_air": "hfc152a",
    "mole_fraction_of_hfc227ea_in_air": "hfc227ea",
```

(continues on next page)

(continued from previous page)

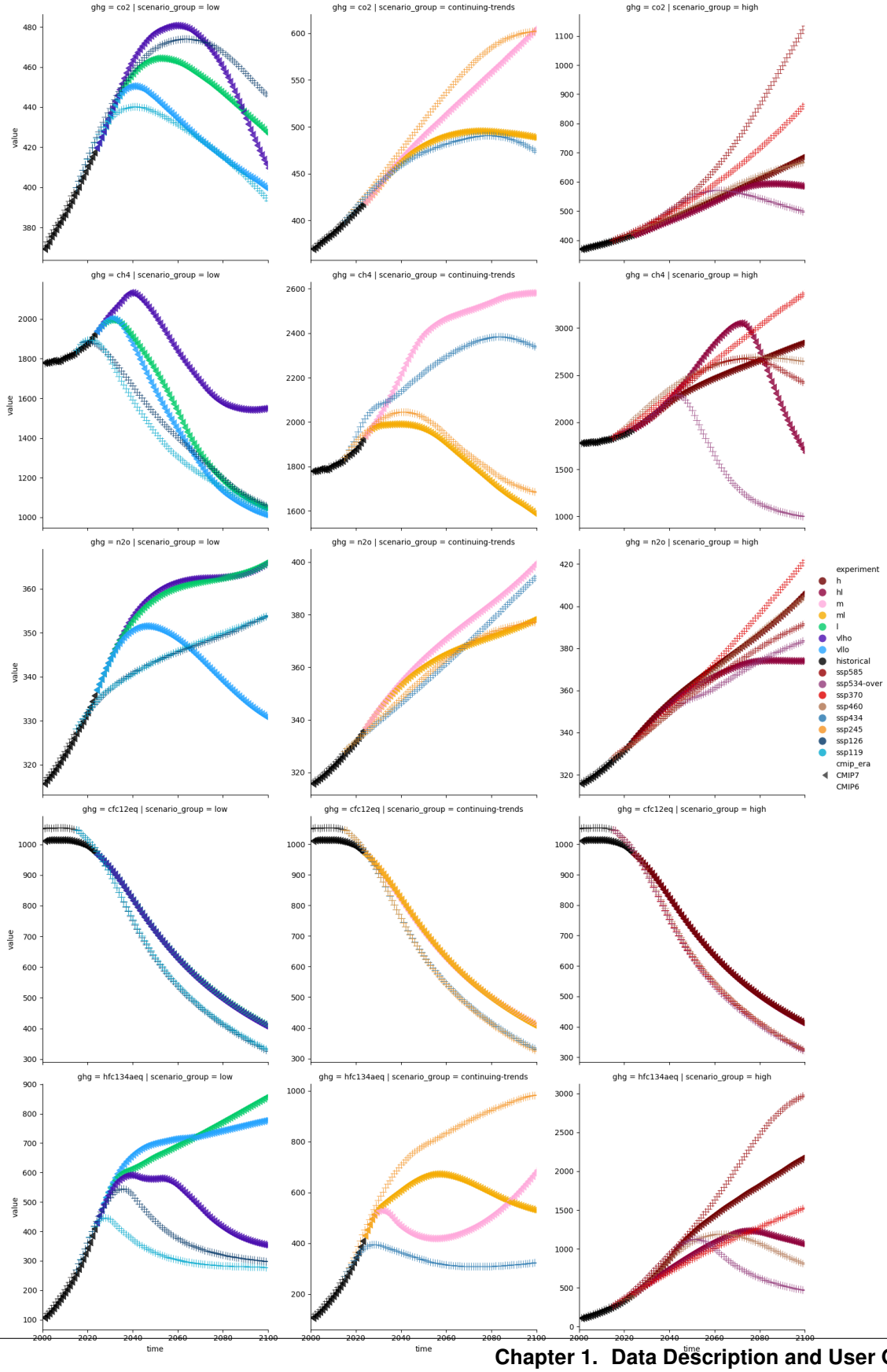
```
    "mole_fraction_of_hfc23_in_air": "hfc23",
    "mole_fraction_of_hfc236fa_in_air": "hfc236fa",
    "mole_fraction_of_hfc245fa_in_air": "hfc245fa",
    "mole_fraction_of_hfc32_in_air": "hfc32",
    "mole_fraction_of_hfc365mfc_in_air": "hfc365mfc",
    "mole_fraction_of_hfc4310mee_in_air": "hfc4310mee",
    "mole_fraction_of_nf3_in_air": "nf3",
    "mole_fraction_of_sf6_in_air": "sf6",
    "mole_fraction_of_so2f2_in_air": "so2f2",
    "mole_fraction_of_cfc11eq_in_air": "cfc11eq",
    "mole_fraction_of_cfc12eq_in_air": "cfc12eq",
    "mole_fraction_of_hfc134aeq_in_air": "hfc134aeq",
}
```

Data comparisons

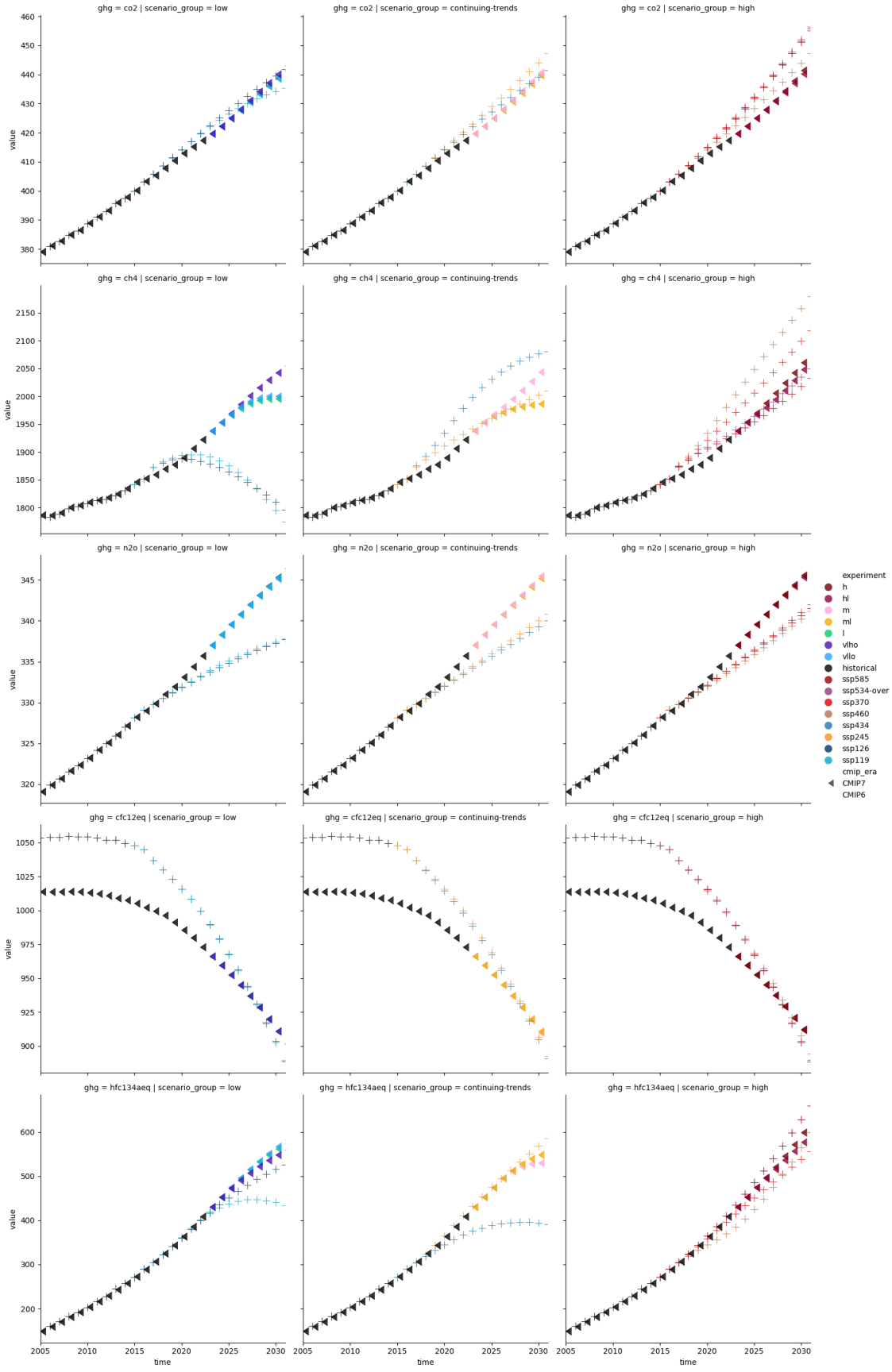
Comparing the data from CMIP6 and CMIP7 shows changes in two key areas:

1. the scenarios are simply different
2. the transition from historical to scenarios is more carefully harmonised

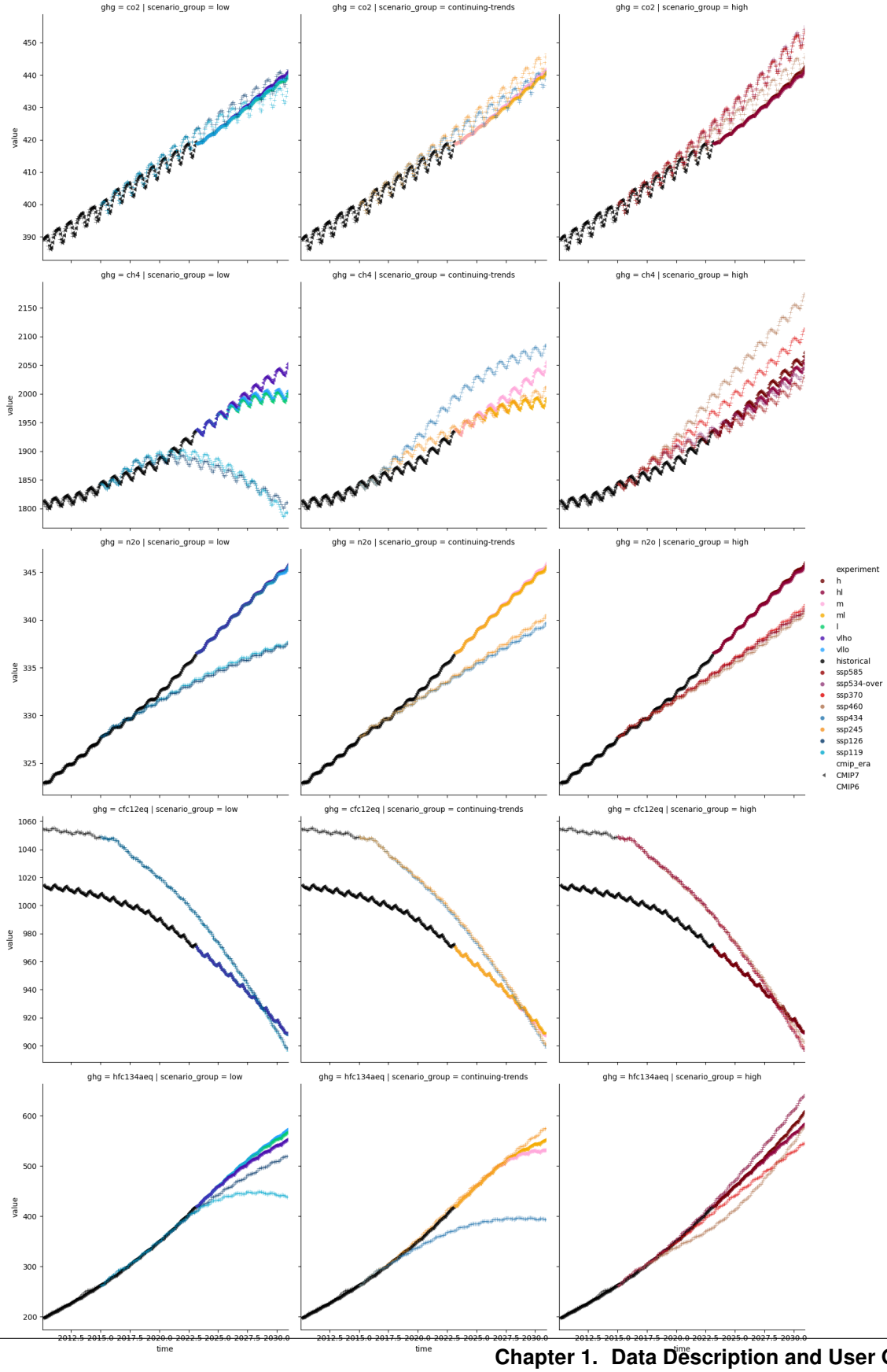
Annual-, global-mean data



If we zoom in, the harmonisation issues in the CMIP6 data are more clear, most obviously for CH4 where the transition from history to scenarios also includes a notable increase in the rate of change that is independent of scenario (i.e. is not driven by a change in emissions).


ghg = co2 | scenario_group = low
ghg = co2 | scenario_group = continuing-trends
ghg = co2 | scenario_group = high
ghg = ch4 | scenario_group = low
ghg = ch4 | scenario_group = continuing-trends
ghg = ch4 | scenario_group = high
ghg = n2o | scenario_group = low
ghg = n2o | scenario_group = continuing-trends
ghg = n2o | scenario_group = high
ghg = cfc12eq | scenario_group = low
ghg = cfc12eq | scenario_group = continuing-trends
ghg = cfc12eq | scenario_group = high
ghg = hfc134aeq | scenario_group = low
ghg = hfc134aeq | scenario_group = continuing-trends
ghg = hfc134aeq | scenario_group = high
value
time
experiment
h
hl
m
ml
l
vlho
vllo
historical
ssp585
ssp534-over
ssp370
ssp460
ssp434
ssp245
ssp126
ssp119
cmip_era
CMIP7
CMIP6

Monthly-, global-mean data



1.6 Bibliography

BIBLIOGRAPHY

[1] Malte Meinshausen, Zebedee R. J. Nicholls, Jared Lewis, Matthew J. Gidden, Elisabeth Vogel, Mandy Freund, Urs Beyerle, Claudia Gessner, Alexander Nauels, Nico Bauer, Josep G. Canadell, John S. Daniel, Andrew John, Paul B. Krummel, Gunnar Luderer, Nicolai Meinshausen, Stephen A. Montzka, Peter J. Rayner, Stefan Reimann, Steven J. Smith, Marten Van Den Berg, Guus J. M. Velders, Martin K. Vollmer, and Ray H. J. Wang. The shared socio-economic pathway (SSP) greenhouse gas concentrations and their extensions to 2500. *Geoscientific Model Development*, 13(8):3571–3605, 2020. doi:10.5194/gmd-13-3571-2020.