

Changes in response to reviewer 1:

Formatted: Heading 1

Response to overview comments

- It could be helpful for users to briefly mention the package infrastructure used by silicone and if and how it is connected to the other packages in the IAMC toolbox (e.g. data transfer using a common csv format, or by API calls to other packages)
 - We have expanded the section discussing the pipeline to mention data transfer. The section now reads:
 - The pipeline is based around the pyam package (Gidden and Huppmann, 2019), specifically its `IamDataFrame` class, which Silicone makes extensive use of. Pyam dataframes easily convert from and to widely-used pandas dataframes, which pyam and Silicone also use internally (McKinney, 2011). The pipeline also includes tools to harmonise (i.e., correct projection made in the past to match now-known emissions) (aneris, (Gidden et al., 2018a) before infilling and to pass the complete projections to climate simulators. The estimation of climatic impact is performed by OpenSCM,
- For me an overview over the processing steps would be great.
 - We describe all the processing steps in the Methods section. Other than getting the data into pyam form and harmonising it (if desired) there aren't any other pre-processing steps needed. We've added a description describing the interaction between silicone and the harmonisation process in the mathematical detail section. Is there something else you would like to see discussed?
 - If the results are to be harmonised, then harmonising both the infiller and target data before infilling is required for improved consistency (otherwise infilling depends on outdated data). Absolute value infilling techniques preserve harmonisation, however ratio-based approaches do not necessarily, and may need harmonisation again afterwards.
- It would be good to have more structure in explaining the different crunchers and infillers (e.g. paragraph headings for each cruncher) The paper will also be used as a reference for silicone and then it's good to easily find information on a specific issue. The information on crunchers and infillers is somehow scattered through the document with bits in the methods section (text and tables) and some aspects in the results section.
 - [Sub-subheadings have been added to the sections detailing the crunchers]
 - [The rank correlations analysis has been added to the results section. Although not formally a result of silicone, this hopefully provides a clearer separation of model description and data.]
- The overview tables are a good idea, but they should be placed in the methods section for the print version of the paper.
 - We agree, but have no control over typesetting – tables were requested to be at the end of the document at this stage. Hopefully the editors will take note!
- It is a bit unclear how robust the results are and how much work is needed to check them before using the results. It is mentioned in some places, especially the tables, that incomplete databases can influence results. For easy use of the package it would be great if the user is warned in these cases. As one purpose of the package is to fill those gaps it would be a problem if the results are unknowingly influenced by the gaps. Maybe you could add a short section on limitations which gives an overview over such possible problems and references the tables for details. See also comments on lines 99-103 and lines 130-133.
 - The code has now been updated so that the default behaviour is to ignore inconsistent data (e.g. scenarios missing values at one time). There is also now a warning when infilling negative lead emissions with the time-dependent ratio.

- [The text for “time-dependent ratio” and “decompose collection with time-dependent” ratio has been changed significantly to reflect this.] “This relies on all scenarios having values for all of these variables, so misses out cases which do not have one of the constituents or only reports at some of the required times, unless the override option “only_consistent_cases” is set to False.”
- The gas basket splitting functionality seems to be not fully developed. The package has the potential to replace the EQW and it’s update used by assessments like the Climate Action Tracker, but in it’s current state it can not yet do that. It would be great if using QRW or EQW with a KyotoGHG constraint was possible for all gases. Could you add information if/how this can be added e.g. as a multiple infiller or a scaling postprocessing step? This would make the package very useful for the climate policy assessment community.
 - Since writing the first draft we made another multiple infiller that does exactly this. We have now included details of it in the paper. Note that one of the gases is infilled by the conservation condition
 - [“Split collection with remainder emissions” entry added to multiple infiller table and flow chart, described the process of infilling differently for breaking up the aggregate values into their components]
- RMS closest: please consider referencing the paper “Warming assessment of the bottom-up Paris Agreement emissions pledges” (YR du Pont, M. Meinshausen. Nature Communications, 2018) where a similar method has been used
 - Good suggestion. Included as follows:
 - The alternative approach of inputting the whole pathway with the smallest mean-squared distance over all time was used in (Robiou du Pont & Meinshausen, 2018). This works well for large databases containing similar paths, but is less useful for smaller databases or for paths with an unusual behaviour over time.
- It is very useful that the authors test the correlation between the different variables. The Spearman’s rank correlation coefficient does not detect nonmonotonic relationships which can be modeled by the Quantile Rolling Window method. Have you tried other methods to detect correlations (e.g. the Hoeffding Dependence Coefficient).
 - As mentioned in the previous correspondence, we have tried only methods that would also not detect non-monotonic trends. Correlations for all variables with CH4 and CO2 were also plotted and inspected by eye for three times and no clear nonmonotonicity was seen, however.
 - We also plotted the relationships between CO2 and all other variables (using the plotting function in the Silicone examples github) to check that there were no obvious cases of a non-monotonic relationship.
- Did you investigate non-emissions lead variables?
 - No, although the mathematics should work identically. We now point out using economic indicators as a potentially use-case for ratio fillers, although strictly speaking it’s not “infilling” if this data is used (it’s just a very basic IAM).
- Have you used the method on other emissions databases (AR5DB, SSPDB, ...). Is this easily possible or does the package need a structure only present in the SR15DB?
 - As described below, we now apply Silicone techniques to infill a model/scenario from AR5. It is also being applied to AR6. Given that all these listed databases use the IAMC format (more or less), they are easily handled by pyam and hence Silicone. The only pinch point for using other databases is getting the data in the pyam structure. Many databases are in pyam or pyam-compliant structures, so following on from the point below we now show infilling a scenario from such a source.
- Finally a more political comment. Why do you feature a fossil fuel company’s scenario in a climate change paper (Shell sky scenario)? This helps their greenwashing attempts. Please provide the scientific explanation for using this scenario.
 - This was originally referenced because that scenario is claimed to be Paris-compliant and was also one the industry scenarios included in the IPCC SR1.5 pathway assessment. However, it is our contention that this

90 depends on the way that the scenario is infilled. However since to demonstrate this point would require careful data handling, harmonisation and running of a climate simulator to show convincingly we will leave that discussion for elsewhere.

- [The Sky model infilling has been replaced by POEM scenario B infilling from AR5. Text and images have been updated to show this.]

95 **Detailed comments – note that line counts are drastically different now**

- Abstract: I think it would be good to mention some of the most important use cases and infilling options in the abstract.
 - Good idea, we have added a sentence
 - We demonstrate the package's utility with three examples: infilling all required gases for a pathway with data for only one emission species, splitting up a Kyoto emissions total into separate gases and complementing a set of idealised emissions curves to provide a complete, consistent emissions portfolio.
- Line 70: It would be helpful for the reader to have short description of the structure of the paper at the end of the introduction.
 - Good idea
 - This paper is structured as follows: the *Methods* section presents an overview of the different infiller methods, then goes through the infiller techniques in precise and mathematical detail. In *Results*, we present our analysis of emissions projections in the SR1.5 database. This includes correlation statistics of the database, and how well Silicone reproduces aspects of it from the rest. We use this to draw conclusions on the implications for using Silicone on unknown data. In *Use Cases*, we present three examples of using Silicone for infilling a pathway with limited information, splitting up an aggregate basket of emissions and infilling stylised emissions trajectories. We end with a summary of our paper.
- Line 83: Please give an overview over the infilling process at the beginning of the methods overview section or by expanding the introduction of the methods section (i.e. what are the steps in the process).
 - This is now included, if you mean this in a philosophical sense. We also now discuss elsewhere the protocol for harmonisation. If you mean in terms of which commands to use in what order, that seems best learnt from code examples, found in the notebooks.
 - Silicone offers a range of tools that apply methods for doing this infilling which are appropriate in different circumstances, depending on the amount of complete data and how much we know about the narrative behind our emissions. These tools are referred to as 'crunchers'. Each of these crunchers takes a 'lead variable', found in both the infiller and target databases, and uses it to infer the value of a 'follower variable', found only in the infiller database (hence missing in the target database). There are also several tools for easily infilling multiple variables, called 'multiple infillers'. These may have multiple follower or lead variables.
 - If the results are to be harmonised, then harmonising both the infiller and target data before infilling is required for improved consistency (otherwise infilling depends on outdated data). Absolute value infilling techniques preserve harmonisation, however ratio-based approaches do not necessarily, and may need harmonisation again afterwards.
- Lines 99-103: Does the code have any flags to control behavior in these cases? Does it warn if this occurs or is it up to the user to control the results?
 - A warning is now reported for ratio methods with negative target leads, although it is up to the user to determine if the use is acceptable or not.

- 135 • Line 119: Where do the estimates averaged over come from? Different historical estimates for the same year, or different years as well? Is it possible to use trends of the last years with data or averages over the last years? If only single last historical data points are used (which is what I understand from table 1) the method is very sensitive to annual fluctuations in emissions data. Thus trending or averaging is needed for robust results that do not drastically change by using one additional year with data that has unusual emissions due to e.g. extreme weather or crises (financial, COVID-19, local crises, ...).

 - 140 ○ The estimates are not averaged over any period. The vast majority (in our case all) of projected emissions trajectories data reported by integrated assessment models is available at 5 or 10 year intervals. Such data points therefore aim to represent systemic changes rather than (sub-)annual effects. Still, in the case of emissions that fluctuate strongly over time, this infiller should be regarded as less reliable. In the manuscript, we describe the potential use case of this infiller, and we now have added this pitfall in Table 1.
- 145 • Line 123: I assume "cases" refers to scenarios and scenarios where the sign for the lead variable does not coincide with the infillee are taken out of the average for both lead and follower variable? Maybe revise language to clarify that. A more general point: it's a nice way to avoid numerical problems at transitions to negative emissions. However, when working on small scenario databases and / or with high temporal resolutions you might run into problems with availability of scenarios that use the same time period for the transition to negative emissions as the infillee scenario. What do you do in those cases? Does the algorithm allow for a fallback option or does it just fail?

 - 150 ○ You assume correctly (rephrased for clarity now). In this case, the infiller hard fails. The user should decide explicitly whether to use the sign-independent version or another infiller.

 - 155 ▪ It will produce an error if there is no data with the required sign.
- 160 • Lines 130-133: As a lot of scenario databases are not fully consistent regarding data completeness (and sometimes sector definitions) it is dangerous to not automatically check for completeness or at least have an optional filter that removes incomplete scenarios.

 - 165 ○ This is indeed somewhat dangerous. We have now changed how this works so that by default, data that lacks either some of the constituent variables or only reports at some of the required times is removed before infilling.

 - 170 ▪ This cruncher is the foundation for the 'decompose collection with time-dependent ratio' multiple infiller. It relies on all scenarios having values for all of these variables, so misses out cases which do not have one of the constituents or only reports at some of the required times, unless the override option "only_consistent_cases" is set to False. It always constructs a new, consistent version of the aggregate variable in case different modellers used different conversion factors in the infiller database.
- 175 • Lines 138: Does "with all the same times" mean data availability for the same points in time? The use of "times" seems unusual and is not intuitive to understand for me.

 - 180 ○ It does, rephrased

 - 185 ▪ The 'RMS closest' cruncher filters the infiller database for models with data at all the times found in the infillee data.
- 190 • Line 142: Line 140 states that the average is taken over all points in time, yet line 142 states that "the value of the lead at one time impacts the whole timeline". That sounds contradictory to me. Can you explain? Also, adding a formula would help.

 - 195 ○ Rephrased, formula added

 - 200 ▪ The 'RMS closest' cruncher filters the infiller database for models with data at all the times found in the infillee data. It then ranks models and scenarios by the root mean squared (RMS) difference between the lead data in the infiller and infillee database, with the average being taken over all timeslices. It returns the follower data from the scenario/model combination with the smallest RMS difference: the formula is $E_f(t) = e_{f,i}(t)$, where the subscript i refers to the model/scenario

case that minimises $\sum_t \left(E_t(t) - e_{l,i}(t) \right)^2$. In the case of a draw, the value that occurs earlier in the infiller database will be used. This is the only cruncher that is not time-independent, i.e. changing the value of the lead at one time may result in different outputs at other times.

- [Many comments about QRW] If I understand correctly, the windows are not windows with clear boundaries but created by the weighting functions. I think the phrasing "Five windows are drawn" is a bit misleading because to me it suggests that the data-points are binned. However, my understanding is that you actually create a smooth distribution from each discrete data point such that at every lead emissions level chosen to calculate a quantile you have all data follower emissions data-points available, just weighted by the weighting function.
 - Mostly correct, one other possible misconception is that we default to 10 windows, 5 was just chosen for illustrative reasons.
 - A number of rolling windows centers (here 5, by default 10) are drawn and a weighting function constructed for each window. It has a continuous distribution, rather than a discrete cutoff, hence the name.
- Please introduce an index for the data points used for the analysis.
 - [We have significantly changed the notation for the formulae to do this.]
- why do you use the approach of in-terpolating between only 5 points. When calculating the quantiles each time you use the method, you could just calculate them for the exact value of the lead emissions. If you pre-calculate the quantiles and use a lookup table to fill emissions (as done in the old EQW), you could use a higher resolution than just 5 points. I get the impression that you kept some aspects of the Generalized Quantile Walk that are no longer necessary with your improved method, because you don't have the binning.
 - The five points we have used are mainly for plotting clarity. It is up to the user how many points they wish to use and they could use arbitrarily high numbers of points (we default to 10 because, in practice, there is little variation at higher resolution than this). Another reason for using fewer points is computational simplicity – calculating a quantile at each point is $O(N)$ in the infiller database, so doing so for each point in the infiller database can get time-consuming for larger infiller databases.
- Line 149: Maybe add "for each time step" to clarify that the operation is carried out at each time step individually. This information could also be added in the description of the process in the following lines.
 - We have added "for each time."
- Line 149: If I understand correctly you don't interpolate between quantiles, but between follower emissions associated to a given quantile at given lead emission values. If that's correct, please rephrase the sentence.
 - That was indeed not the interpolation we were trying to talk about here
 - infills the values based on interpolating between the quantiles of the follower variable.
- Line 155: There is an additional full stop.
 - [At some points, the whole text of the caption was used rather than just a reference. This has been deleted]
- Line 162: The formula uses r_i as sum index, but neither r_i nor r_p are introduced in the text. (I assume r_i are the lead emission data points in the scenario database ordered by the associated follower value and r_p is the highest one to be taken into account at quantile q (so should it not be r_q instead of r_p ?))
 - [Significantly rewritten in accordance with the above]
 - $q(e_i(j)) = \sum_{e_f(i) < e_f(j)} w_p(e_i(i)) + \frac{w_p(e_i(j))}{2}$.
- Line 181: Why did you choose 0.03?
 - To select only extreme cases.

- We also calculate the variation of this value with time, and in cases where this exceeds 0.03 (chosen to highlight only extreme cases),
- Line 215: Did you analyze if the RMS closest cruncher chooses scenarios coming from the same model or study such that chosen pathways are very similar?
 - We didn't in detail, however they typically this won't happen as the same model will have a large spread in well-modelled variables between the scenarios by construction (there's little point in repeating yourself). By definition, the RMS closest cruncher will choose similar pathways in terms of trend.
- Line 252: The sentence starting with "that" is incomplete.
 - Again, a figure-caption error
 - [deleted]
- Line 256: Does the EQW use interpolation between quantiles and smoothing before calculation of the quantiles?
 - We have added have a section explaining this properly, as follows
 - The equal quantile walk calculates the quantile of the lead value at each time. This 0 for values below the database minimum, 1 for those above the database maximum and the fraction of infiller data smaller or equal to this value otherwise. We interpolate between neighbouring values in the infiller data to avoid rounding errors.
- Lines 259-262: Did you test the influence of the weighting function on the conservativeness of the QRW cruncher. I assume that in the extreme case of constant weighting functions this would always return the median of the scenario database. So the smoothing through the weighting function has to be used carefully.
 - We did, you are correct.
 - Increasing the decay length will reduce the weight difference between points, so the rolling window becomes wider and more even, with the limit case of calculating quantile q of all data for large d_t .
- Line 275: See also comment on line 123. How many scenarios are left when restricting to scenarios with same sign. This will especially be problematic for extreme stylized scenarios with early negative emissions.
 - Indeed, see comments above.
- Line 279: a similar approach has been taken in "Warming assessment of the bottom-up Paris Agreement emissions pledges" (YR du Pont, M. Meinshausen. Nature Communications, 2018)
 - [Reference added]

Comments on testing the examples repository

- There are problems with directory names. The input data directory in the repository is called "input", while in the code is referenced as "Input" with upper case "I". This leads to the code failing on unix systems.
 - [Fixed]
- Splitting up basket: The notebook fails in cell 7, line 8 with: "TypeError: convert_unit() missing 1 required positional argument: 'to'"
 - This problem should not happen with the latest version of silicone, do you still find it?
- Stylized Path: the legend of the plot resulting from cell 9 shows 4 cases of MESSAGE scenarios, yet in the plot only two can be distinguished. Is that correct?
 - Correct, E_∞ does not appear in this formula so two of these lines are identical.

- Is it possible to store calculated quantiles for the QRW and EQW method to increase calculation speed by using a lookup table?
 - The quantiles for QRW are calculated at the point of generating the infiller, so several different target databases can be infilled efficiently in this way, but it doesn't speed up our use case here. More lookup tables are possible in principal but this would not demonstrate the use of this toolkit in practice though. We could add such functionality in future but we don't believe it is necessary to illustrate the tool in this paper (pull requests and issues at the github repository are most welcome). We have reduced the computational load of the new POEM infilling notebook relative to the Shell notebook by removing most of the variables, which were never plotted or explored anyway.

Comments on tables and figures

- Table1: latest time ratio: see comment on line 119. Please explain what re-harmonizing means in this context.
 - [Comment has been removed here]
- Table1: Time dependent ratio: when using the optional filtering for same sign, how do you ensure that there is a sufficient number of scenarios available?
 - We raise an error if there are none of the correct sign, It's not clear that there's a "sufficient number" required above 1. There will now be a warning thrown if the target data is negative, irrespective of the number of negative infiller scenarios.
- Table 1: Linear interpolation: Is it possible to use scenario binning here to not select two very similar pathways (e.g. same model and storyline with small parameter variation)? I think this could be made more stable by using more than two scenarios, but the problem will not vanish. Can you give an example, where using this cruncher makes sense?
 - This cruncher is only intended for use on a very limited number of scenarios. We tend to want cases where the scenarios have similar model and storylines. This then ensures that the infilled result is compatible with those storylines. E.g. my model assumes a SSP5 world and is comparable to these groups of scenarios with high biomass burning – I want to infill the amount of BC corresponding to this much CO2. We also now mention that this is very similar to a previously developed interpolator that used cubic spline interpolation, which is fairly similar in the limit of a large database. This specific cruncher in Silicone was recently used in the paper *Current and future global climate impacts resulting from COVID-19* <https://www.nature.com/articles/s41558-020-0883-0>
 - A tool for infilling was provided with (Rogelj et al., 2014) using a cubic spline between specific points in a small database, however this type of infiller behaves chaotically when applied to large databases incorporating many different models. It was also coded in Excel, limiting the ease of open-source development.
- Table1: EQW: Does the EQW cruncher use smoothing through a weighting function when calculating the quantiles?
 - EQW is smooth but does not have a weighting function
 - [Section on EQW added elsewhere]
- Tables 1 and 2: please repeat the table header after a page break
 - Good suggestion. This would be a matter for typesetters. Tables do not behave properly with track changes so we will not attempt to do this by hand here
- Fig 2b: there are some artifacts around the subfigures (partly visible plot boundaries?)
 - [hopefully fixed now]
- Fig 3: The spike in the linear interpolation pathway shows that this cruncher is very problematic.

- Correct and intentional. This is partly alleviated in the current version, but we do not recommend using linear interpolation without filtering the data down carefully, as is described in the text.
 - We see from Figure 3 that the linear interpolation model (without filtering the database) provides a chaotic pathway, due to its value being determined only by the two points either side of it in the database, which changes semi-randomly with time and should not be used here.
- Figs 4-6: please add a legend
 - Added legends to the figures.

Changes in response to reviewer 2:

Changes made

1a. In most parts of the manuscript and in the source code the specified aim is to complete missing emissions in future scenarios produced by IAMs. However, in some sections of the manuscript, other aims are indicated (e.g. complete stylised scenarios, fill missing sectoral data, use historical estimates, aggregate regional data(?)) which I found confusing sometimes.

One solution could be to remove these hints of other applications and use the (IAM/stylized) future scenarios as the aim and example throughout the paper. Additional applications could then be outlined in a discussions section (in a bit more detail).

- To reduce the confusion, we have removed the reference to aggregating regional data entirely and have rephrased the section around historic estimates and sector emissions. Since one of the crunchers (last time ratio) is specifically designed to deal with the case of historic emissions, it is still discussed there. We mention regions in the context of structure as follows: “Pyam dataframes assign values to variables as a function of different models, scenarios, regions and times. All methods work on databases with only a single region at a time, although the region can be different between the infiller and target databases.”

1b. With respect to the main aim I would have expected use cases showing the completion of several variables of different IAM scenarios.

- As mentioned in the separate discussion piece submitted previously, we are somewhat confused by this comment – the first “use case” provides a case where a scenario (now a different scenario) has many variables completed, the second and third cases complete variables from multiple scenarios. There is also a github repository consisting only of examples of use (some of which are mentioned in this paper, some not – note that the Sky scenario example in the text has changed now) and a “notebook” section to the main github talking through the general principles of applying these techniques. Could you clarify if there is something else you would like to see added, and whether you really want this included in the paper itself? (We are essentially treating the examples github as a living and evolving Supplementary Info, although the archived version is always available from Zenodo.)

2. I would recommend to rework parts of the abstract and of the introduction in order to better cover the content/ set the scene for the rest of the manuscript (e.g. add info about rank correlation, tests/ use cases).

Formatted: Heading 1

- We now mention the use-cases in the abstract and have an outline of the manuscript in the introduction. “A variety of infilling options are outlined and their suitability for different cases are discussed. We recommend certain infilling techniques as the good defaults, but emphasise that considering the specifics of the model being infilled will produce better results. We demonstrate the package’s utility with three examples: infilling all required gases for a pathway with data for only one emission species, splitting up a Kyoto emissions total into separate gases and complementing a set of idealised emissions curves to provide a complete, consistent emissions portfolio.”

3. Literature: it is rather difficult for me to imagine that there are no other somewhat comparable tools around and that so far missing emissions were usually set to zero or only somehow unsystematically filled following ‘ad hoc’ decisions (as stated in the introduction). For completeness it would be nice if the authors could dig some more into the literature and check how climate models so far got the required input from IAMs? One example for a tool covering a similar purpose in maybe a slightly different but connected setting is the tool used in Gütschow et al. (2016) and du Pont et al. (2016) which is described in Nabel et al. (2011). Some of the co-authors have been involved in these papers.

- As discussed with Robert Gieseke in previous correspondence, some specific papers you mention do not strictly do infilling – they use non-emissions economic data to perform the analysis, whereas we deem ‘infilling’ to be when no such data is available.
 - We have changed the tense of the sentence to clarify that explicitly setting these values to zero is rare, although very often done implicitly by ignoring the variable altogether. In explicit cases, the Equal Quantile Walk is used. We now mention Nabel et al as an example of using this but note that there is no general tool for this so we think our open-source codebase provides a significant step forward in terms of reducing duplicated effort. We also now highlight the use of a RMS-closest technique by Robiou du Pont & Meinshausen, 2018. The text now reads:
 - “Most earlier studies overcame this problem in one of two ways: with expert-based ad-hoc decisions on how to adequately fill-in missing species (Schaeffer et al., 2015); or by assuming that a pathway will occur at the same quantile for each set of emissions in a particular year, although the quantile can vary over time (Gütschow et al., 2018; Meinshausen et al., 2006; Nabel et al., 2011). However, the former clearly does not scale easily to larger databases (because making ad-hoc decisions for a thousand scenarios requires a significant time input), and the latter approach, termed the “Equal quantile walk” (EQW), ignores trade-offs and specific relationships between emission species resulting from how competing technologies, behaviours and industrial practices result in different emissions. A few alternative approaches have been used recently: for instance, using the pathway with the smallest mean-squared distance over all time was used in (Robiou du Pont and Meinshausen, 2018). This works well for large databases containing similar paths, but is less reliable for smaller databases or for paths with an unusual behaviour over time. A more sophisticated “Generalized Quantile Walk” technique can capture the effect of trade-offs and was recently introduced in section 3.8.1 in (Teske et al., 2019), involving quantile regression between a lead variable (fossil CO₂ emissions) and other gases for every individual year. Unfortunately, the implementation there did not consistently guarantee that higher quantiles resulted in higher emissions, and has not been followed up with any peer-reviewed work that does so. A tool for infilling was provided with (Rogelj et al., 2014) using a cubic spline between specific points in a small database, however this type of infiller behaves chaotically when applied to large databases incorporating many different models. It was also coded in Excel, limiting the ease of open-source development.”

4. While reading I sometimes got confused by different terms and I found parts of the manuscript a bit sketchy or difficult to read. More specific:

a) There are several changes in terminology among different (sub-) sections (e.g. “lead variable”, “lead gas” “inputs or outputs” (l.89), and timeseries; and a sudden switch to model and scenario in 2.2.2 where 2.2.1 only had the more general term database; but also small things as the change from CH₄ in section 1 and 2 to methane starting section 3). I think for the reader it would be helpful to stick to a certain terminology throughout the manuscript.

- References to “lead/follower gas” have been changed to “lead/follower variable”. Inputs is now generally removed where it is equivalent to “infiller”, see below. Timeseries refers to any data that changes over time and is used to refer to any of the above – it is a data structure, not a description of the data meaning.
 - A short paragraph explaining what model/scenario combinations are has been added before 2.2.1: “As one final detail, we discuss the data model which is assumed by Silicone. Silicone is built around the pyam package (Gidden and Huppmann, 2019). As a result, it assumes that all input data is provided in a particular structure. The structure includes the model which created the timeseries, the scenario with which the timeseries is associated (e.g. a high BECS 1.5 degree scenario), the region the emissions occurs in and the unit of the data (full details available at <https://pyam-iamc.readthedocs.io/en/stable/data.html>). Accordingly, Silicone is able to work on specific subsets of models (e.g. only the MESSAGE model) or subsets of scenarios (e.g. all SSP1-like scenarios). We therefore follow the pyam convention and refer to a “model/scenario combination” to mean a single projected world, that in some contexts might be called a “scenario”.”

b) The terms infiller and infilly are very difficult to distinguish in quick reading and I think it would help a lot when choosing less similar terms – How about source and sink/target, or infiller and target, or infiller and silicon-filled, or comparable.

- Infiller and target seems like a good combination, now used throughout.

c) In subsection 2.2.2, a bit out of the sudden, several scenarios and models seem to be presupposed, while at the beginning of section 2 only “a database that contains data for at least two emission species” is kind of officially introduced. Maybe it would help to directly introduce the use of different models/IAMs and their scenarios at the beginning of section 2 such that the usage of different timeseries (2.2) and different models and scenarios (2.2.2) is less out of the sudden. An alternative could be a consistent use of the more general terms of “timeseries of different lead variables in the infiller database” depending on the main aim of the paper (see also point 1).

- Change made as described in 4a), introducing the structure of pyam dataframes.

d) Please consider to better structure 2.2.1 and 2.2.2 (e.g. print algorithm names in bold or in italics, with separated paragraphs for the different algorithms or /and as lists (e.g. latex ‘description’ or the like)).

- Sub-subtitles now used, in bold.

e) Equations are throughout embedded in the text (see also point 5).

- Key equations now in “display” mode.

5. Since the different algorithms for the completion of emission timeseries are the main scientific contribution of the paper I would appreciate if the equations could be clearly separated from the text (i.e. introduced as separated numbered equations) and, furthermore, if more equations would be added (see also specific comments below). In

430 my opinion this could increase readability (4e) and reproducibility. Ideally it could also help to better understand how cases of several lead variable pathways are treated in the different algorithms.

- As above. Important equations are numbered for external reference and we have added several equations for clarity.

6. Test and use cases only show the usage of absolute value based algorithms, which

435 I find unfortunate and a bit incomplete given the share of the method section dedicated to the ratio based algorithms. How about at least including examples using the time dependent ratio method?

- One of the ways of splitting the Kyoto gas totals is the “decompose collection with time-dependent ratio”, which is a wrapper calling the ratio-based method several times. We have modified the text to make this clearer that this such an example.

440 7. Please check the format of your references in the text (e.g. I.28, I.59, I.60, I.68, : : :)

- This has been corrected.

445 **Specific comments/questions**

I.1 Why Silicone?

- Our package fills what some would call gaps in emissions scenarios. Silicone is a caulking agent used to fill in gaps in tiling and was the first thing one of the developers thought of when we searched for a name. We are happy to add an explanation in the text if you feel it is necessary.

450 I.16 Transition. E.g. “In this paper: : :”

- “This paper presents a variety of infilling options and outlines their suitability for different cases are discussed. We recommend certain infilling techniques as the good defaults, but emphasise that considering the specifics of the model being infilled will produce better results. We demonstrate the package’s utility with three examples: infilling all required gases for a pathway with data for only one emission species, splitting up a Kyoto emissions total into separate gases and complementing a set of idealised emissions curves to provide a complete, consistent emissions portfolio.”

455 I.16 Please consider to add more information here about the content of the paper

- Detail added, as above

I.33 : : : exert : : : between? Please check language

460 ○ Changed to “as a large number of supposedly minor emissions may collectively exert a significant radiative forcing.”

I.36 Is there an example reference/study where filling with zeros has been applied?

- Implicitly, this is done by every study where any of the F-gases are ignored (e.g. every IAM in the SR1.5 database ignores NF₃). However, studies don’t tend to list the gases they ignore.

465 ○ “If no infilling is attempted, the unevaluated emissions would effectively be considered zero, which would clearly create systematic biases and potential artefacts in the projected temperatures.”

I.41 I do not understand “does not scale easily”

- Added “However, the former clearly does not scale easily to larger databases (because making ad-hoc decisions for a thousand scenarios requires a significant time input)”

470 I.45 There is no 3.8.1 in this reference (reference currently points to Chapter 4, if you target 3.8.1 consider citing Teske et al.)

- Corrected as advised

I.60 Please specify what “this” refers to

- Now “Silicone”

475 I.64 “suite of tools” are these all python tools?

- o Indeed, “Python” added

I.66 Are there more than these two? Else consider to add “and/as well as” before “harmonise...”

- The pipeline can be considered to include the OpenSCM part, although this isn’t strictly developed by the IAMC, it’s therefore a little fuzzy.

- o “This pipeline includes tools to manipulate and plot IAM data (pyam, (Gidden & Huppmann, 2019)) and harmonise mismatches in historical emissions (aneris, (Gidden, Fujimori, et al., 2018)). The estimation of climatic impact is performed by OpenSCM, managed by the OpenSCM community (Nicholls, Gieseke, Lewis, & Willner, 2020), which is compatible with the data structure of the pipeline.”

I.67 Consider deleting “, managed by the OpenSCM community”

- As above, we have revised the sentence slightly according to this comment, while the text still makes clear that this is not part of the IAMC

I.82 several? Three/two: ::

- Currently 4 (one due to an update from the original paper draft), more may well emerge. One is arguably an “aggregation tool” rather than a “multiple infiller”. We maintain ambiguity for this reason

I.85 Maybe change to “Currently, there are ...”?

- o “Currently” added

I.98 Consider rephrasing e.g. “... and where emissions are expected to scale with each other ...”

- o Rephrased, this discussion is moved to the next paragraph. “The ratio-based approaches are better for cases where the lead values to be infilled are outside the range in the infiller database and we expect the emissions to scale with each other.”

I.98 What do you mean with “regional data” and “aggregate data”? If you refer to regions as subset of global data then this is the first time that a spatial reference is given and I wonder if it would be appropriate to introduce this more formally earlier in the manuscript?

- o Reference to regional data has been removed as discussed above. A reference to “splitting up an aggregate basket of emissions” mentioned above, this case now changed to “splitting up aggregated emissions into their components”

I.100 Consider to give an example? CO2 uptake?

- o Added “, e.g. CO₂ emissions”

I.106 What do you mean with similar – similar magnitudes?

- o Indeed, clarified to “, preferably with both larger and smaller lead emissions in the infiller database”

I.111 “estimate the ratios” – if not to be predefined: ::

- o Now “determine”

I.112 “follower value in infillee database” -> “in the ...”

- o Added as requested

I.112 Please consider to visually separate (and number) the equations.

- o Change made for all significant equations

I.114 “each different timeseries” -> different regarding what - do you mean different follower variables?

- o Sentence removed.

I.116 mean regarding what - time or different sources (models, scenarios?)

- o Sources, clarified to “The ‘latest time ratio’ method uses the ratio between the mean follower data in the infiller database (we denote this database with lower-case, e_f) and the value of the lead variable in the target data (E_t)”

I.117 what does “both” refer to?

- Both means follow and target lead. Now “both values”

I.119 what do you mean with “all estimates”? Different sources (models, scenarios)?

- Yes, specification of “at that time” added

525 I.120 why historical? Couldn’t this also be different scenarios from different IAMs?

- It can, it’s simply that this is the most common use-case (we are yet to see a case where all IAMs model a gas up to a given point then all stop thereafter whilst continuing to model other gases).

I.120 “and the lower case ef (t) represents the follower values in the database at time t.” -> “and the lower case ef (tlast) represents the follower values in the database.”?

530 ○ Rephrased as requested, the distinction being between *E* and *e*.

I.125 the infilllee lead is not part of the formula – is this the EI(t) from I.113?

- Yes, it will be used to multiply the ratio.

I. 125-133: Maybe consider to restructure? You could start with the context, i.e. the algorithm name, and then add the explanation, e.g.: “The decompose collection multiple

535 infiller is based on : : relying on the useful property of : : :”

- Significantly restructured and added more equations as explanation

I.134-137: Equation for R(t)?

- We explain that this follows the logic below instead, it would be tedious to write out here again with one symbol different. The whole passage has been rewritten.

540 I.142 I did not understand this sentence

- Rephrased to “This is the only infiller that is not time-independent, i.e. changing the value of the lead at one time may result in different outputs at other times.”

I.143-145: Equation for Ef?

- We have added it.

545 I.151 There is no 3.8.1 in that reference (reference currently points to Chapter 4, if you target 3.8.1 consider citing Teske et al.)

- Well spotted, resolved as above

I.152 There seems to be a lost copy of the figure caption in the text. (Either just delete or maybe rephrase to steps with complete sentences)

550 ○ Indeed, deleted

I.155-165 Equation for Ef?

- This would be illegibly long and complex without a great deal of separately defined objects. The whole section has been rewritten however.

I.163 Maybe Ef not EI?

555 • The weights are associated with e_f , the ordering is associated with e_l . This has been rewritten, as noted above.

I.166 What about the KyotoGHGs as one basket?

- Could be considered (and can be calculated by the code in the examples github repository if you’re interested), but interpreting it is complicated by the fact that
- 0) calculating the Kyoto GHGs as a single basket only makes sense if all relevant Kyoto gases are reported, which isn’t often the case (if it were Silicone’s domain of applicability would be severely reduced)
- 1) there are several possible metrics for this, on top of which the values stored in the IIASA database for a given metric do not entirely equal the values calculated from their components. Sometimes this is due to the incompleteness issue, other times there appear to have been rounding errors or method disputes in the process.
- 2) it naturally correlates very well with its primary components CO2 and methane without really signifying anything, further complicating the averages analysis.

565

I.169 Maybe give an example for two such variables?

- Added “for instance black carbon and carbon monoxide are both produced by incomplete combustion”

I.185 Consider explicitly listing the two constituents

- OK, it is somewhat long-winded to spell out the acronyms here, so we simply use them and hope those unfamiliar with them will look at the table
- “(AFOLU and Energy and Industrial processes, a similar concern can be raised about F-gases)”

I.186 Which are “these two” – CO₂ and CH₄?!

- Yes, spelt out now. “CO₂ and CH₄”

I.188 you write that BC, CO and OC “correlate poorly with others, however, from the table it seems that they do not correlate less well with others than other gasses, the main difference is that they correlate very well with each other.

- Clarified: “that correlate well with each other but less well with other emission pathways”

I.190 maybe aggregate F-gas emissions / F-gasses as a basket?

- As a basket added to the analysis

I.191 up to here always “CH₄”

- Changed to always be CH₄

I.198 consider deleting “and find similar results”

- Ok

I.199 “we choose four” – which are basically all?

- Following from the point below, this now reads: “We use the crunchers that are designed for use on complete datasets with only default settings: QRW (default settings mean in absolute mode and for the 0.5 quantile), RMS closest, EQW, time-dependent ratio and linear interpolation. Interpolate selected model behaves identically to linear interpolation with default settings and is not treated separately here.”

I.199 But even if there are errors, wouldn’t it be interesting to see what happens? I

would appreciate if you could also show results for the ‘time dependent ratio’ algorithm

- We now show the results from using the time-dependent ratio too, and slightly change our normalisation to make the comparison with “just use the mean” clearer. In several cases it does indeed give values worse than 1 (i.e. using the mean). A complicating consequence of this new normalisation and higher possible values is a strong skewing of the results and non-normal distribution of the errors. We therefore also substitute the Wilcoxon t-test for the student t-test in all cases. The statistics are all robust to this and it has no impact on our conclusions, although all of the p-values are slightly different now. References to Time-dependent ratio have also been added to the conclusion here.

I.201-202 Equation?

- The (new) equation has been added: “i.e. $\sqrt{\left(\frac{E_{f,inf}-E_{f,act}}{\sigma}\right)^2}_{i|decade}$, with the subscript text *inf* indicating that the value is infilled, *act* indicating actual and *i/decade* indicating averaging over model/scenario cases or decades.”

I.203 “both cases” – CO₂ and CH₄?

- Indeed, clarified

I.203 “non-CO₂ pathways” – but CO₂ is derived with CH₄ -> maybe replace by “emission pathways”

- We have replaced this with “follower pathways”

I.204 I would rephrase this, if QRW would be fairly similar all four would be?

- Rephrased as “the next smallest”

I.208 What do you conclude from the non-Gaussian distribution test?

- 610
- Originally we concluded that we could use the Student’s t-test. With the new normalisation and data, we now fail this test so use the Wilcoxon t-test. The qualitative conclusions are as before.
- I.212 “either of CO2 or CH4”
- Corrected to “either CO₂ or CH₄”
- I.214 capital T for Table 3
- 615
 - Corrected
- I.225 Add Silicon -> "Data in the Silicon package"
- Clarified to “in the Silicone examples package”, the main Silicone package is not dependent on any fixed data source. (Other data sources are used by the examples package, but are also included in it.)
- I.238 treatment of regions has not been introduced, maybe explain better or consider
- 620 deletion?
- Regions are now mentioned above, but not here, as described above.
- I.243 “this database” – which? The SR1.5 repository?
- This whole section has changed, the data here now comes from a different (named) database, AR5.
- I.249-252: Again a lost copy of a figure caption
- 625
 - Deleted
- I.285-287 “free variables” are mentioned twice but are not further explained?
- They are not tremendously relevant to the discussion here, but this has been changed to “in this case, based on rates of transition between the RCP pathways and a long-term emissions value”
- I.287-288 I do not understand this sentence
- 630
 - Changed to “Silicone provides an alternative means of complementing such results – instead of specifying the functional forms of all emissions, you can have a few key emissions prescribed and infill the remainder using scenarios with similarities to the desired narrative.”
- I.300 consider deleting “of which there are many” or maybe replace by “several options”
- or the like
- 635
 - Removed
- Table2: Please explain the asterisk again in this figure caption
- Added “Names followed by asterisks use a ratio-based approach, i.e. they find a multiplicative factor and then multiply the target lead by this value, if the asterisk is in brackets there are ratio-based options.”
- Table3: Consider to change the colouring – to me the yellow/orange highlighting gives
- 640 a ‘positive’ impression. Maybe you could colour the cells with bold numbers in green
- and those which are currently yellow in red?
- The table has been recoloured. We have avoided using green and red for colourblindness reasons, but yellow for strong correlation and blue for high time-variability should perform the same roles.

Formatted: Normal

645

Silicone v1.0.0: an open-source Python package for inferring missing emissions data for climate change research

Robin D. Lamboll¹, Zebedee R. J. Nicholls^{2,3}, Jarmo S. Kikstra⁴, Malte Meinshausen^{2,3}, Joeri Rogelj¹

¹Grantham Institute for Climate Change and the Environment, Imperial College London, UK

² Australian-German Climate and Energy College, The University of Melbourne, Parkville, Victoria, Australia

³ School of Earth Sciences, The University of Melbourne, Parkville, Victoria, Australia

⁴ International Institute for Applied Systems Analysis, Laxenburg, Austria

Correspondence to: Robin Lamboll (r.lamboll@imperial.ac.uk)

Abstract. Integrated assessment models (IAMs) project future anthropogenic emissions which can be used as for-input into for climate models. However, the full list of climate-relevant emissions is lengthy and most IAMs do not model all of them. Here we present **sSilicone**, an open-source Python package which infers anthropogenic emissions of missing-unmodelled species based on other known-reported emissions projections. For example, it can infer nitrous oxide emissions in one scenario based on carbon dioxide emissions from that scenario plus the relationship between nitrous oxide and carbon dioxide emissions found in other scenarios. ThisSilicone's infilling capability broadens the range of IAMs available for exploring projections of future climate change. Silicone and forms part of the open-source pipeline for assessments of the climate implications of IAM scenarios, led by the Integrated Assessment Modelling consortium Consortium (IAMC). This paper presents. Aa variety of infilling options are outlined and outlines their suitability for different cases are discussed. We recommend certain infilling techniques as good defaults, but emphasise that considering the specifics of the model being infilled will produce better results. We demonstrate the package's utility with three examples: to-infilling all required gases for a pathway with data for only-for one emission species, to-splitting up a Kyoto emissions total into separate gases and-to complementing a set of idealised emissions curves to provide a complete, consistent emissions portfolio. The code and notebooks explaining details of the package and how to use it are available from-theon GitHub repository, (<https://github.com/GranthamImperial/silicone>). There-is-an-additional repository showing-with this paper's examples and uses of the code to complement existing research is available at https://github.com/GranthamImperial/silicone_examples.

<Note to reviewers: the version of Silicone in the bibliography is not the finalised version v1.0.0 to allow for corrections before a v1.0.0 release.>

1. Introduction

1.1. General context and problem setting

Integrated assessment models (IAMs) are scientific modelling tools that integrate knowledge from different academic disciplines with the aim to explore and inform policy decisions (Clarke et al., 2014; Rogelj et al., 2018a; Weyant, 2017).

They are widely used in climate change research to combine insights from energy, economy, agricultural, and natural sciences, with the aim ~~to create~~of creating scenarios that explore how societal decisions can affect projected greenhouse gases and other emissions, as well as their related climate outcomes (Clarke et al., 2014; Huppmann et al., 2018; Riahi et al., 2017; Rogelj et al., 2018b).

However, IAMs do not always exhaustively represent all possible processes or sources of climate-relevant emissions. Thus, many IAM scenarios lack projections for some climate forcers, be it specific greenhouse gas emissions or aerosol precursors. A complete set of these climate forcers is required to accurately estimate the overall climatic effects of a given scenario (Meinshausen et al., 2011; Smith et al., 2018), as a large number of supposedly minor emissions may collectively exert a significant radiative forcing ~~between them~~ (Meinshausen et al., 2017; O'Neill et al., 2016).

Scenarios that only report a limited set of greenhouse gases or climate forcers thus must be complemented by estimated evolutions of missing emissions derived without further economic analysis. We term this estimation 'infilling'. If no infilling is attempted, the unevaluated emissions ~~would be~~ would be effectively ~~be infilled with~~ be ~~considered~~ considered zeros, which ~~would~~ clearly create systematic biases and potential artefacts in the projected temperatures. Depending on the radiative forcing of the species in question, this bias may be positive or negative, so infilling with zeros would not necessarily be a conservative choice. Most earlier studies overcame this problem in one of two ways: with expert-based ad-hoc decisions on how to adequately fill-in missing species (Schaeffer et al., 2015); or by assuming that a pathway will occur at the same quantile for each set of emissions in a particular year, although the quantile can vary over time (Gütschow et al., 2018; Meinshausen et al., 2006; Nabel et al., 2011). However, the former clearly does not scale easily to larger databases (because making ad-hoc decisions for a thousand scenarios requires a significant time input), and the latter approach, termed the "Equal quantile walk" (EQW), ignores trade-offs and specific relationships between emission species resulting from how competing technologies, behaviours and industrial practices result in different emissions. A few alternative approaches have been used recently: for instance, using the pathway with the smallest mean-squared distance over all time was used in (Robiou du Pont and Meinshausen, 2018). This works well for large databases containing similar paths, but is less reliable for smaller databases or for paths with an unusual behaviour over time. A more sophisticated "Generalized Quantile Walk" technique can capture the effect of trade-offs and was recently introduced in section 3.8.1 in (Teske et al., 2019), involving quantile regression between a lead variable (fossil ~~CO₂~~CO₂ emissions) and other gases for every individual year. Unfortunately, the implementation there did not consistently guarantee that higher quantiles resulted in higher emissions, and has not been followed up with any peer-reviewed work that does so. A tool for infilling was provided with (Rogelj et al., 2014) using a cubic spline between specific points in a small database, however this type of infiller behaves chaotically when applied to large databases incorporating many different models. It was also coded in Excel, limiting the ease of open-source development.

Here we present a new toolbox of methods to address these recurring infilling challenges in the climatic assessment of socioeconomic emissions scenarios. The toolbox introduces new approaches as well as building on and combining -previous

710 approaches. ~~The codebase is a significant improvement on compared to existing options~~~~It significantly improves the~~
~~codebase~~ in terms of flexibility, applicability, reproducibility and versatility.

1.2. The aim of Silicone

715 Silicone is a Python package designed to enable users to expand scenario projections of a limited set of climate forcers to a
~~full-broader~~ set required for a ~~sensible~~~~accurate~~ climate~~ie~~ assessment. In essence, its methods are grounded in a comparison
of the co-evolution of anthropogenic emissions in scenarios that are readily available in the literature (Huppmann et al.,
2018; Riahi et al., 2017; Rogelj et al., 2018b). Silicone aims to provide IAM teams that do not represent all individual
climate forcers with robust methods to complement their model output and facilitate a climatic assessment of their work.
Furthermore, Silicone also aims to provide geoscience researchers with a tool to easily develop stylized, yet internally
consistent future emission pathways of the most important climate forcers. It can also estimate or calculate missing
720 ~~emissions from particular sectoral datasets~~. Notebooks describing how to use these tools are available on the
accompanying GitHub repository (Lamboll et al., 2020b) and the formal documentation is available at (Lamboll et al.,
2020a). Additional examples of using ~~this Silicone~~ for the specific situations outlined below are included in a separate
GitHub repository (Lamboll, 2020). The package is open-source and intended to allow groups to write their own infilling
methods if desired. Users and collaborators are ~~invited to submit their code for review by the team~~~~encouraged to add any~~
725 ~~such developments to the codebase via~~ on GitHub.

Silicone is compatible with a suite of Python tools that make up the IAM climate assessment pipeline developed under the
umbrella of the Integrated Assessment Modelling Consortium (IAMC). ~~The compatibility with these tools allows us to load,~~
~~manipulate and save files using a common file format. This~~ pipeline ~~is based around the pyam package~~ (Gidden and
Huppmann, 2019), ~~specifically its IamDataFrame class, which Silicone makes extensive use of. Pyam dataframes easily~~
730 ~~convert from and to widely-used pandas dataframes, which pyam and Silicone also use internally~~ (McKinney, 2011). ~~The~~
~~pipeline also includes tools to manipulate and plot IAM data (pyam, (Gidden & Huppmann, 2019)), to harmonise (i.e.,~~
~~correct projection made in the past to match now-known emissions) (aneris, (Gidden et al., 2018a)) before infilling and to~~
~~pass the complete projections to climate simulators~~. The estimation of climatic impact is performed by OpenSCM, ~~managed~~
~~by the OpenSCM community~~ (Nicholls et al., 2020), which is compatible with the data structure of the pipeline. This
735 pipeline is being developed in support of the IAM community and the IAM scenario assessment for the forthcoming Sixth
Assessment Report of the Intergovernmental Panel on Climate Change (IPCC AR6) in particular.

~~This paper is structured as follows: the Methods sections presents an overview of the different infiller methods, then goes~~
~~through the infiller techniques in precise and mathematical detail. In Results, we present our analysis of emissions~~
~~projections in the SR1.5 database. This includes correlation statistics of the database, and how well Silicone reproduces~~
740 ~~aspects of it from the rest. We use this to draw conclusions on the implications for using Silicone on unknown data. In Use~~

Formatted: Font: Italic

Formatted: Font: Italic

Formatted: Font: Italic

Cases, we present three examples of using Silicone for infilling a pathway with limited information, splitting up an aggregate basket of emissions and infilling stylised emissions trajectories. We end with a summary of our paper.

2. Methods

Silicone takes a database that contains data for at least two emissions species (this database is referred to as the ‘infiller’ database) and derives a relationship between these timeseries. It then applies that relationship to a second database (the ‘infiller+target’ database), which does not have any data for one of the emissions species in the infiller database. For example, based on an infiller database of CO₂ and N₂O emissions, Silicone could then derive N₂O emissions compatible with the CO₂ emissions in a less complete infiller+target data-base. In all cases, the infillers will perform best if the target data comes from a scenario that is socioeconomically similar to scenarios found in the infiller database. The performance of most crunchers can be improved by filtering out scenarios that are known to assume radically different characteristics like population number before infilling, provided that comparable emissions statistics can be found in the remaining database.

Silicone offers a range of tools that apply methods for doing this infilling which are appropriate in different circumstances, depending on the amount of input-complete data and how much we know about the narrative behind our emissions. These tools are referred to as ‘crunchers’. Each of these crunchers takes a ‘lead variable’, found in both the infiller and infiller+target databases, and uses it to infer the value of a ‘follower variable’, found only in the infiller database (hence missing in the infiller+target database). There are also several tools for easily infilling multiple variables, called ‘multiple infillers’. These may have multiple follower or lead variables.

2.1. Methods overview

2.1.1. Cruncher guide

There are currently seven types of cruncher. These are outlined in Table 1 below. A flow chart to guide the choice is shown in Figure 1. There is also a series of notebooks with examples of how to use them all in the main GitHub repository (Lamboll et al., 2020b).

2.1.2. Multiple Infiller and Aggregation Tools Guide

Multiple infillers are for cases where there are relationships between multiple lead or follower values for inputs or outputs values that need to be considered at the same time. They allow less tailored approaches to infilling but can ensure that the infilling is faster or more consistent than infilling each of the variables separately. These are outlined in Table 2.

2.2. Mathematical detail

Notebooks presenting benefits and risks of each cruncher type can be found in the ~~s~~Silicone GitHub (Lamboll et al., 2020b) and may be useful to have as examples when analysing the work below, as well as demonstrating how to use them.

770 There are two main classes of infillers: those based on the ratios between two emission pathways and those based on the absolute emission values in the infiller database. If the results are to be harmonised, then harmonising both the infiller and target data before infilling is required for improved consistency (otherwise infilling depends on outdated data). Absolute value infilling techniques preserve harmonisation, however ratio-based approaches do not necessarily, and may need harmonisation again afterwards.

775 The ratio-based approaches are better for cases where the lead values to be infilled are outside the range in the infiller database and we expect the emissions to scale ~~together with each other~~. For instance, if we are infilling one incomplete combustion product regional data from aggregate databased on another, or splitting up aggregated emissions into their components ~~gases~~. However, care needs to be taken when infilling emissions that are non-negative using a lead value that may be of any sign, e.g. CO₂ emissions. In that case, the ratio method might produce values for the ~~infilleetarget~~ emissions that are unsupported by any available evidence. Singular behaviour may also be encountered when the ~~average~~ lead data is close to zero in the infiller database. The different crunchers present different ways to estimate the ratio to use.

780 The absolute value-based techniques infill with values derived from the absolute data found in the infiller database, or linear combinations of them. This means that they will always return values within the range spanned by the infiller database. This is most appropriate for processes where we have a ~~larger-greater~~ number of cases, preferably with both larger and smaller similar lead emissions in the infiller database or where we expect the follower emissions to be strongly bounded rather than increasing in line with other variables. They may be considered more stable and more conservative. The quantile rolling windows (QRW) cruncher can be used in either ratio or absolute (non-ratio) mode, the absolute mode being the default.

785 As one final detail, we discuss the data model which is assumed by Silicone. Silicone is built around the pyam package (Gidden and Huppmann, 2019). As a result, it assumes that all input data is provided in a particular structure. The structure includes the model which created the timeseries, the scenario with which the timeseries is associated (e.g. a high BECS 1.5 degree scenario), the region the emissions occurs in and the unit of the data (full details available at <https://pyam-iamc.readthedocs.io/en/stable/data.html>). Accordingly, Silicone is able to work on specific subsets of models (e.g. only the MESSAGE model) or subsets of scenarios (e.g. all SSP1-like scenarios). We therefore follow the pyam convention and refer to a “model/scenario combination” to mean a single projected world, that in some contexts might be called a “scenario”.

795 Pyam dataframes assign values to variables as a function of different models, scenarios, regions and times. All methods work on databases with only a single region at a time, although the region can be different between the infiller and target databases.

Formatted: Subscript

Field Code Changed

Formatted: Font: 10 pt

2.2.1. Ratio infilling methods

These methods all firstly estimate the ratio of the lead ~~gas-variable~~ to the follower-~~gas variable~~. In all cases, we first ~~determine estimate~~ the ratios, written as $R(t)$ at time t . Once these have been calculated, the follower value in ~~the infilltarget~~ database, $E_f(t)$, is valued as

$$E_f(t) = R(t)E_l(t), \tag{1}$$

where $E_l(t)$ is the lead value in the ~~infilltarget~~ database.

Constant ratio and latest time ratio crunchers

‘Constant ratio’ and ‘latest time ratio’ methods both use the same ratio for all infill times, $R(t) = R$. ~~The ratios may still be different for each different timeseries in the infill database.~~ With the ‘constant ratio’ method, the ratio must be given as an input parameter. The ‘latest time ratio’ method uses the ratio between the mean follower data in the infiller database (we denote this database with lower-case, e_f) and the value of ~~the~~ lead variable in the ~~infilltarget~~ data (E_l), both ~~values~~ evaluated at the latest time for which there exists follower data in the infiller database, t_{last} . ~~The mean is taken over all infiller data at that time. This is designed for the case where we have estimates only up until some time assumes that we have historic data up until sometime, after which it stops – for instance, if we have no projections for some new HFC emissions, but have historic measurements for recent years.~~ This gives us the equation

$$R = \frac{\langle e_f(t_{last}) \rangle}{E_l(t_{last})}, \tag{2}$$

where the angular brackets mean taking the (algebraic) mean with equal weighting for all estimates (typically historical estimates) ~~at that time,~~ and ~~the~~ with a lower case, $e_f(t)$ represents the follower values in the database at time t . This ensures that at t_{last} , all infilled data will ~~infill~~

$$E_f(t_{last}) = R \cdot E_l(t_{last}) = \langle e_f(t_{last}) \rangle.$$

Time-dependent ratio cruncher

The ‘time-dependent ratio’ ~~is appropriate for method relies when there is on having some~~ data in the infiller database for all times, and allows the ratio to vary with time. ~~For this The ratio used we use is~~

$$R(t) = \frac{\langle e_f(t) \rangle}{\langle e_l(t) \rangle} \quad (3)$$

Optionally, the averaging can be taken only over model/scenario cases where the sign of the lead variable is the same in both the infiller and infiller/target case – this will guarantee that the infilled value takes the same sign as that of follower values in the database. It will produce an error if there is no data with the required sign. This cruncher has a A-useful conservativity property of this relation (with or without the sign restriction): is that if in every scenario averaged over, the emissions of several substances sum to another substance, e.g. if $e_1 = e_2 + e_3$, then $\langle e_1 \rangle = \langle e_2 \rangle + \langle e_3 \rangle$. It then follows that and so

$$1 = \frac{\langle e_2 \rangle}{\langle e_1 \rangle} + \frac{\langle e_3 \rangle}{\langle e_1 \rangle} \quad (4)$$

the right-hand side of which we can identify as the two $R(t)$ values of using formula (3) twice for different followers. This means when the aggregate is the lead and the components are followers, the sum of the two ratios is one, so we can use this infiller to break an aggregate value into components and know that the total is conserved. This relationship generalises to any number of components, still holds when emissions can be negative, and is irrespective of whether the averaging includes all values or only those where the lead has a particular sign.

This cruncher is the foundation for the ‘decompose collection with time-dependent ratio’ multiple infiller. It does however rely on all scenarios having values for all of these variables, so misses out cases which do not have one of the constituents or only reports at some of the required times, unless the override option “only consistent cases” is set to False. It always constructs a new, consistent version of the aggregate variable in case different modellers used different conversion factors there is an inconsistency or absent data in the infiller database. This is not performed by the cruncher, so missing or inconsistent data can potentially distort results here.

Quantile rolling windows cruncher

The ‘quantile rolling window’ method may be applied in ratio mode, in which case we calculate $R(t)$ by first calculating the ratio for each scenario,

$$r_s(t) = \frac{e_f(t)}{e_l(t)} \quad (5)$$

then following the calculation in the absolute value section, using this instead of e_l . This method finds quantiles of the ratio in the infiller database at set points along the range of lead values in the infiller database. It is more fully described below in the absolute value section.

Formatted: Heading 3

855 2.2.2. Absolute value infilling methods

2.2.2. RMS closest cruncher

The ‘RMS closest’ cruncher filters the infiller database for models with data at all the same-times found in the infilleetarget database. It then ranks models and scenarios by the root mean squared (RMS) difference between the lead data in the infiller and infilleetarget database, with the average being taken over all timeslices. It returns the follower data from the scenario/model combination with the smallest RMS difference: the formula is $E_f(t) = e_{f,i}(t)$, where the subscript i refers to the model/scenario case that minimises

$$\sum_t \left(E_l(t) - e_{f,i}(t) \right)^2. \quad (6)$$

In the case of a draw, the value that occurs earlier in the infiller database will be used. This is the only infiller that is not time-independent, i.e. changing the value of the lead at one time may result in different outputs at impacts the whole timeline other times.

Linear interpolation

The ‘linear interpolation’ method constructs an (unsmoothed) linear interpolator function between all lead and follower points in the infiller database at a given point in time. It is similar in concept to the cubic spline interpolator used in (Rogelj et al., 2014). The equation for our case is

$$E_f(t) = \max_{t < t} e_{f,t}(t) + \min(E_l(t) - e_{l,t}(t)) \left(\frac{e_{f,t}(t) - e_{f,t}(t)}{e_{l,t}(t) - e_{l,t}(t)} \right), \quad (7)$$

where subscript $<$ or $>$ signs indicate the model/scenario combination with lead values immediately below or above the target lead value at that time. If multiple points have exactly the same lead value, the average follow value is used. The follower value returned is then the interpolated value for the infilleetarget lead. The ‘Interpolate specified scenarios and models’ cruncher filters for scenarios and models that match a given text string before performing the same action of the linear interpolation cruncher.

Quantile rolling windows cruncher

The ‘quantile rolling windows’ cruncher, applied with the default option ‘use_ratio=False’, infills the values based on interpolating between the required quantile of the follower variable. This is calculated at fixed points across the range of lead values in the infiller database for each time. This process is identical to the above discussion where ‘use_ratio’ is =True, except using the actual follower values instead of the ratios between lead and follow. It is inspired by the Generalized Quantile Walk approach in section 3.8.1 of (Meinshausen and Dooley, 2019). An illustration of the idea behind this cruncher

Formatted: No bullets or numbering

Formatted: Heading 3

is shown in Figure 2: Schematic of how the quantile-rolling-window cruncher determines the follow value to use. a) Example relationships between lead (CO_2) and follow (CH_4) variables over time. b) Five windows are drawn and a weighting function constructed for each window. c) A relationship between the sum of the weights and the follow value is established and the follow value at the desired quantile is returned. Figure 2. For each time in the infiller database, it splits the range of lead values into n_{windows} points (defaults to 10) with values e_p , including the highest and lowest values. For each window, the weightings of each point are given as

$$w_p(e_l(i)) = 1 / (1 + \left(\frac{(e_p - e_l(i))}{d_l} \right)^2), \quad (8)$$

where d_l is the decay length, which defaults to half the separation between e_p , and i the label for which model/scenario we are investigating. Increasing the decay length will reduce the weight difference between points, so the rolling window becomes wider and more even, with the limit case of calculating quantile q of all data for large d_l . Amongst other things, this is a clear improvement over the Generalized Quantile Walk approach, as the latter uses equal weights within a fixed window of a certain fraction of the infiller database's lead values in a certain year. These values are then normalised so that $\sum w_p = 1$ and sorted into ascending order by e_f . The follow value at quantile q , evaluated at lead point $e_{lp}(j)$, is where the quantile equals the sum of weights of all smaller e_f plus half the weight of $e_f(j)$ itself. Note that we sum over smaller follower values, but the weighting is determined by the lead values:

$$q(e_l(j)) = \sum_{e_{fp}(i) < e_{fp}(j)} w_p(e_{fi}(i)) + \frac{w_p(e_{fp}(j))}{2}. \quad (9)$$

Quantiles between these these known points are evaluated by linearly interpolating this relationship. We are usually interested in the case where $q = 0.5$. To infill a point at E_l , we interpolate between the known points e_p . Quantile crossing is not possible in this framework because at any given evaluation point higher quantiles cannot have lower values, and only linear fits between these points are used.

Equal quantile walk

The equal quantile walk calculates the quantile of the lead value at each time (Meinshausen et al., 2006). This is zero for values below the database minimum, one for those above the database maximum and the fraction of infiller data smaller or equal to this value otherwise. We interpolate between neighbouring values in the infiller data to find the fraction that would match the target value exactly. We then apply the same logic to calculate the appropriate value for the derived quantile of the follower data.

Formatted: Font: Italic

Formatted: Font: Italic

Formatted: Font: Italic

Formatted: Heading 3

3. Results

3.3.1. Rank Correlations

The infilling method is important. However, equally important is the choice of lead variable. The best choice is where there is a causal link between the lead and follower variable, particularly if there is a clear understanding of the implications of this link for the relative behaviour of the two variables, for instance black carbon and carbon monoxide are both produced by incomplete combustion. In most cases, there is no such certainty, and the best choice is then to find the lead variable with the best predictive power. We estimate this by the Spearman's rank correlation coefficient, a measurement of the monotonicity of the relationship between the two variables. In cases where this value is low, we anticipate the need for higher effort to select relevant cases from the infilling database. We use the data from the IPCC Special Report on Global Warming of 1.5 °C (Huppmann et al., 2018) as our database of scenarios and compare the correlations between the different variables. The Silicone package has a function in the statistics section called 'calc_all_emissions_correlations', which will produce tables of both the Spearman (rank) and Pearson correlation coefficients, calculated separately for each year requested and also the time-averaged magnitude of the correlations. Since there is no reason to expect the relationships between variables to be linear, we will focus on the rank correlation in this analysis. We also plotted the relationships between CO₂ and all other variables (using the plotting function in the Silicone examples github) to check that there were no obvious cases of a non-monotonic relationship. All the crunchers work just as well with negative trends as with positive, so the sign of the correlations is not relevant for considering goodness of fit. Using this tool, we can calculate the decadal-averaged magnitude of the rank correlation coefficient, found in Table 3.- We also calculate the variation of this value with time, and in cases where this exceeds 0.03 (chosen to highlight only extreme cases), colour the cells orangeblue. This is to indicate cases where more care needs to be taken to ensure that values are representative for the times of interest.

The immediate observation from the study of absolute rank correlations is that there is no clear, overall best infiller gas. CH₄ has a slightly higher average than other emissions and is reported by most models. CO₂ is reported by all models and has the second highest correlation, however this is somewhat inflated by having two of its constituents listed separately (AFOLU and Energy and Industrial processes, a similar concern can be raised about F-gases). Generally, these two CO₂ and CH₄ therefore are therefore the best choices for a 'default lead variable'. However, there are some specific cases where the correlations are low, and much better choices could be made.

There is a cluster of values-emissions species, specifically black carbon, organic carbon and carbon monoxide BC, CO, and OC, that correlate poorly well with each other but less well with other emission pathways. Physically, these relate to incomplete burning, and are best infilled using each other. The F-gases, SF₆, hydrofluorocarbons (HFCs), and perfluorinated compounds (PFCs) also primarily relate to each other. Fortunately, mMany models report F-gas emissions as a basket, so,

Formatted

Formatted: Subscript

Formatted: Subscript

Formatted: Subscript

~~infilling~~ Infilling these should best be done by splitting the F-gas basket into its constituents. Otherwise the default infillers, CO₂ and CH₄ methane, should do reasonably.

4. Results

4.1.3.2. Reconstructing data

945 The choice of cruncher to use in different situations will depend on the expectations about the specific emissions in question. However, in cases where there are no clear expectations, it is good to have a default. In this section we assess to which degree the cruncher reproduces the follower data from one model and scenario given the lead data from that case and all data from all the other model/scenario combinations in the SR1.5 database. We try this with both CH₄ methane and CO₂ as our lead variables ~~and find similar results. Since we have no reason to believe that the answers will always scale with these variables (and ratio-based infillers have potentially unbounded errors), we~~ use the crunchers that are designed for use on complete datasets with only default settings ~~choose four absolute value infillers as our options: QRW (using only default settings mean, so in absolute mode and for the 50th 0.5 quantile), RMS closest, EQW, time-dependent ratio and linear interpolation. Interpolate selected model behaves identically to linear interpolation with default settings and is not treated separately here.~~ We perform the infilling for each model/scenario combination, for each decade from 2020 to 2100, and report the root mean squared difference between the original value and the infilled value, normalised by the standard deviation ~~the total variation~~ in the follower value in the database at that time (σ), i.e. $\sqrt{\langle \left(\frac{E_{f,inf} - E_{f,act}}{\sigma} \right)^2 \rangle_i}$ ~~with the subscript text *inf* indicating that the value is infilled, *act* indicating actual and *i/decade* indicating averaging over model/scenario cases or decades.~~ These results are found in tables 4 and 5. Given the definition of standard deviations, values larger than one would indicate that this infiller is worse than simply using the mean value in the database.

960 We see ~~that with this fairly large infiller database that in both for both CO₂ and CH₄ methane with this fairly large infiller database,~~ the approach that generates non-CO₂ follower pathways most similar to those removed from the initial scenarios (i.e. the smallest errors) is the RMS technique, ~~with and that the the~~ QRW technique ~~being is the next smallest~~ fairly similar. Linear interpolation without smoothing is expected to produce a noisy fit when given a large ~~input infiller~~ dataset, so its performance is unsurprisingly worse. The Equal Quantile Walk (EQW) performs similarly poorly, due to effectively ignoring the relationship between the lead and follower data. The time-dependent ratio method is worst of all – its errors are potentially unbounded and for CO₂ the average error far exceeds one. To determine the appropriate statistics to apply on the errors, we first perform a Shapiro-Wilks test to detect any non-Gaussian aspect for the error distribution (details can be found in the “statistics for paper” notebook of the examples github repository). This indicates that neither CO₂ nor the distributions methane are statistically significantly non-Gaussian in distribution, for several crunchers either when analysed separately when analysed separately and most clearly for each cruncher or as an aggregate. We will therefore use non-

970

Formatted: Normal, No bullets or numbering

Formatted: Superscript

Formatted: Subscript

Formatted: Subscript

Formatted: Subscript

Formatted: Subscript

parametric tests where possible. The small differences in rank between ~~CH₄methane~~ and CO₂ manifest in slightly lower values for ~~CH₄methane~~. Performing a ~~student's-Wilcoxon's~~ t-test on the results indicates that this result is statistically significant for the data as a whole (relative t-test t-statistic ~~3763.4~~, p = 0.000079043), although when considering each of the crunchers individually, only the RMS closest ~~and time-dependent ratio~~ crunchers ~~is~~are significantly better with ~~CH₄methane~~ than CO₂ (p-values for ~~time dependent ratio = 0.012~~, QRW = 0.3348, RMS Closest = 0.09841, linear interpolation = 0.07860, EQW = 0.379). We therefore conclude that using either ~~CO₂ or CH₄~~ as the default will produce ~~the most~~ reasonable results ~~when using one infiller species~~, with ~~this slight preference for using CH₄methane performing slightly better, while also generally having a as our default infiller counteracted by the~~ slightly lower availability of data ~~(412 cases rather than 414, as seen in Table 3)~~.

We perform similar pairwise ~~Wilcoxon~~ t-tests on the results of different crunchers, and find that the ordering of mean errors, (RMS closest < QRW < Linear interpolation ≈ EQW < ~~Time dependent ratio~~) are all statistically robust. The p-values are < 0.0014 for ~~almost~~ all pairs except linear interpolation and EQW, which are much greater than 0.1, whether the comparison uses ~~CO₂ lead data~~, or ~~CH₄methane~~ lead data, or ~~both all data combined for all cases~~. ~~The one exception to this is time-dependent ratio and EQW for CH₄, which has only p=0.028, though the values for other combinations still have p < 0.01.~~

We stress that this does not always mean that the RMS closest technique is the best default, as it makes the assumption that the pathway being infilled is similar to a whole pathway found in the database. The advantage of the quantile rolling windows technique is its ~~choice of~~ conservativity – ~~e.g.~~ that it tends to produce values more towards the median value ~~if the default 0.5 quantile is used~~ – and time-independence, whereas RMS closest is better at reconstructing the data and has better consistency over time. Linear interpolation, ~~and~~ EQW ~~and time-dependent ratio~~ are best used in cases where there is a large degree of knowledge about the expected relationship between variables.

4.2.4. Use cases

Data in the ~~Silicone examples~~ package relies on the ~~IAM Consortium~~ (IAMC) ‘pyam’ open-source software data structure (Gidden and Huppmann, 2019) and fits into the IAMC scenario assessment pipeline prepared in support of the IPCC ~~AR6~~ literature assessment.

As part of the pipeline, emissions projections are also harmonised, i.e. modified to be consistent with known historical emissions in a smooth way (Gidden et al., 2018a). The Silicone process is assumed to be part of the IAMC pipeline after harmonisation, as the harmonisation process will potentially differently affect the ~~infilled target~~ and infiller data, resulting in inconsistencies. All infiller options except latest time ratio are designed such that if both the data being infilled (the ‘~~infilleetarget~~ data’) and the data drawn on for infilling (‘infiller data’) are harmonised, the result must also be harmonised, so there is no need for harmonisation again after infilling. (Latest time ratio only preserves the harmonisation of the last

Formatted: Subscript

Formatted: Subscript

Formatted: Subscript

Formatted

timepoint in the infiller database.) The infilled results can then be run via climate models, most easily via the OpenSCM package (Nicholls et al., 2020).

We now demonstrate several uses of the package for specific purposes. The notebooks demonstrating the steps for these calculations can be found in the *[S-silicone_examples](#)* GitHub repository (Lamboll, 2020), *along with several other use-cases*. All methods work on databases with only a single region at a time, although the region can be different between the infiller and infillee databases. In our case, we always use global data.

4.2.1.4.1.1. Infilling the Shell IMAGE model POEM Sky-scenario B

To demonstrate the uses of this package alone, we will apply the methods directly using unharmonised data in the SR1.5 repository (Huppmann et al., 2019) to infill the emission pathways of the *POEM scenario B Sky-scenario from the AR5 database* (Clarke et al., 2014) *by the Shell World Energy Model (which is also found in this database)*. The *Sky-POEM* scenarios only reports *CO₂ from certain sources a limited set of greenhouse gases* and *is are* thus an excellent use case. The crunchers are all used via the multiple infiller, “infill_all_required_emissions_for_openscm”. No active decisions are taken except to use the *SSP25* scenarios from the *REMIND MAgPIE 1.5 MESSAGE* model for the specified model interpolation. The choice of *SSP25* in this case is ultimately arbitrary but supported by *is due to the Sky-POEM scenario B being fairly middle-of-the-road and usually fitting in this the SSP2 range, involving similar assumptions of high-industry, high-development, high-energy demand and high fossil fuel use.* The choice of *MESSAGE* model is because this is the marker model for *SSP2 that scenario* (Riahi et al., 2017). *Other POEM scenarios would need different ranges of scenarios for infilling.*

We see from *Figure 3* *Figure 3: Left: The Shell Sky model CO₂ from Industry and Industrial Applications data. The fine lines represent the different timeseries in the SR1.5 database used to perform the infilling and are not included in the legend for clarity. Right: The results of interpolating this data using four different crunchers. The interpolate specified model approach used the REMIND MAgPIE 1.5 model and only choses scenarios based on SSP5 pathways.* that the linear interpolation model (without filtering the database) provides a chaotic pathway, due to its value being determined only by the two points either side of it in the database, which changes *semi-randomly with time and should not be used here*. Although the interpolate specified model approach is also determined by only a few model/scenario pairs because there is only data from a small number of related scenarios, the pathway is smoother and more consistent. The EQW pathway *assumes a strong, direct relationship between looks somewhat jagged, due to the density of pathways around the Shell sky model. This means that small differences in CO₂ emissions can result in reasonably large changes in quantile and hence infilled CO₂ and CH₄ methane emissions which the other crunchers do not uphold at early times, although this would disappear if the data were harmonised.* The other cruncher results are all fairly similar and look consistent. The RMS closest pathway is consistent by construction (and precisely overlines a point in the original database). The quantile rolling windows

Formatted: Font: Italic

Formatted: Subscript

Formatted: Subscript

result also looks consistent and ~~tends to move is noticeably~~ closer to the dense clouds of values in the infiller database around 200 Mt CH₄ than other crunchers, which ignore data that is not highly similar in some way. In deciding which is the best infiller to use, the RMS closest result is more consistent over time but more arbitrary in its selection of the pathway, while quantile rolling windows is more conservative in the sense of giving results closer to the median behaviour of the whole data set.

4.1.2. Splitting up a Kyoto Greenhouse Gases path

The Silicone package has features that can split a basket of gases into its constituents. In this example we take data from the Climate Action Tracker (CAT) website (<https://climateactiontracker.org/>) (Climate Action Tracker, n.d.), which reports projected global emissions in terms of Kyoto gas totals (Jeffery et al., 2018). While it is possible to use this to infill all other values directly as above, the subcategories of Kyoto gas will not necessarily add up to the Kyoto gas total, ~~one of the multiple infillers designed for this use is preferable. so a preferable method is~~ The symmetric way to divide the basket into its constituent parts (CO₂, CH₄, N₂O and F-gases), ~~is using the first using the~~ 'decompose collection with time-dependent ratio' multiple infiller, ~~which uses a ratio-based technique to ensure conservation of the total amounts. Alternatively, the 'split collection with remainder' multiple infiller then infill the remaining gases using other techniques as before can estimate the fractions of CH₄, N₂O and F-gases, then assign the remainder to CO₂.~~ F-gases could be further subdivided using similar methods.

As can be seen in Figure 5, the curves that result from decompose collection are generally smooth, in spite of being separately calculated at each timepoint. It is important to ensure that the number of scenarios reported at each time are consistent. In the SR1.5 database, some scenarios only report values at decadal intervals, whereas others use five-year intervals. We interpolated all models to five-year intervals to give consistent representation. In the CH₄ and F-gases, the lowest orange line is clearly seen to rise discontinuously after 2060. This is the last point before the Kyoto total goes negative. To ensure that the sign of the constituents is correct, the formula only considers data from SR1.5 paths where the Kyoto total has the same sign as in the data being infilled. In this way, emissions that are unlikely to go negative like CH₄ methane are ensured positive, however their magnitude increases the more negative the aggregate is.

For this reason, the 'split collection with remainder' method produces more robust results with sign changes in the lead variable. An alternative way to infill, which is preferable in this case where we know that only one of the emissions will go negative. This technique can use is to use any cruncher, usually RMS closest or (probably non-ratio) quantile rolling windows to infill the positive values and then allow the value that may be negative (CO₂) to make up the rest. This produces the results seen in Figure 6. Here behaviour of all curves is fairly smooth, with no obvious features around zero-crossing points and no negative values except in CO₂, as expected.

Formatted: Outline numbered + Level: 3 + Numbering Style: 1, 2, 3, ... + Start at: 1 + Alignment: Left + Aligned at: 0 cm + Indent at: 0.89 cm

Formatted: Subscript

4.2.2.4.1.3. Stylised trajectories

Another use of this software is to infill simple, stylised trajectories generated to explore a wide range of possibilities without detailed economic modelling. For example, Sanderson *et al.* (Sanderson et al., 2016) suggest simple formulae whereby one may construct emissions trajectories characterised by a few free variables – in this case, based on rates of transition between the RCP pathways and a long-term emissions value. They present general formulae for generating plausible total CO₂ pathways with several free variables. Silicone provides an alternative means of complementing such results – instead of specifying the functional forms of all emissions, you can have a few key emissions prescribed and by-infill the remainder using scenarios with similarities to the desired narrative, ~~removing the need for more numerical free parameters between~~ formulae. A notebook can be found in the Silicone examples github detailing the calculations and demonstrating this usage, titled “Infill_stylised_path.ipynb” (Lamboll, 2020), using data from (Riahi et al., 2011; van Vuuren et al., 2011). It shows that curves with different values in some of the parameters, termed E_{∞} and τ , can be complemented using a number of techniques. Here we highlight the method of interpolating results from any of the SSP scenarios as implemented by variants of the MESSAGE models. As the different SSPs ~~models-scenarios~~ have different narratives, this allows the user to decide what narrative is relevant to the infilling, rather than adding more arbitrary values (Gidden et al., 2018b). An example of this output can be found in Figure 7.

5. Summary

In this paper we have outlined the features of the open-source Silicone package. This provides tools for complementing emissions pathways with other climate-relevant emissions through relationships found in the scenario literature. The package features several scripts for analysing data to establish the relationships between the variables in the complete infiller database, to establish the best variables to use when infilling. The values of the follower data are estimated using objects called crunchers, ~~of which there are many~~. Notebooks describing the use of the crunchers are included in the a GitHub repository (<https://github.com/GranthamImperial/silicone>), which also contains full documentation, ~~and~~. In addition, a flowchart for ~~choosing the best~~ guiding the choice of cruncher for ~~your a given~~ situation is included in the text. The results of Spearman’s rank correlations and applying the crunchers to the SR1.5 database implied that, ~~in cases of complete ignorance~~, the best ~~default guess at~~ lead variables are CH₄ methane and CO₂, and that the best default cruncher is the root mean squared closest cruncher, followed by the quantile rolling windows cruncher. Both of these crunchers perform significantly better at reconstructing known pathways compared to the currently commonly used equal quantile walk technique, although this and many other crunchers are included in the package for specific situations where they are more appropriate ~~situationally specific usage~~ convenience. Using several examples and use-cases of different ~~Several use-cases of the~~ infilling techniques,

this paper has ~~are~~ demonstrated that Silicone can easily be used to allow a broader range of IAMs to make reasonable climate assessments.

Bibliography

- Clarke, L., Jiang, K., Akimoto, K., Babiker, M., Blanford, G., Fisher-Vanden, K., Hourcade, J.-C., Krey, V., Kriegler, E.,
1095 Löschel, A., McCollum, D., Paltsev, S., Rose, S., Shukla, P. R., Tavoni, M., Zwaan, B. van der and Vuuren, D. van:
Assessing Transformation Pathways, in Climate Change 2014: Mitigation of Climate Change. Contribution of Working
Group III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change, edited by O. Edenhofer, R.
Pichs-Madruga, Y. Sokona, E. Farahani, S. Kadner, K. Seyboth, A. Adler, I. Baum, S. Brunner, P. Eickemeier, B. Kriemann,
J. Savolainen, S. Schlömer, C. von Stechow, T. Zwickel, and J. C. Minx, pp. 413–510, Cambridge University Press., 2014.
- 1100 Climate Action Tracker: Climate Action Tracker, [online] Available from: <https://climateactiontracker.org/> (Accessed 9 July
2020), n.d.
- Gidden, M. J. and Huppmann, D.: pyam: a Python Package for the Analysis and Visualization of Models of the Interaction
of Climate, Human, and Environmental Systems, J. Open Source Softw., doi:10.21105/joss.01095, 2019.
- Gidden, M. J., Fujimori, S., van den Berg, M., Klein, D., Smith, S. J., van Vuuren, D. P. and Riahi, K.: A methodology and
1105 implementation of automated emissions harmonization for use in Integrated Assessment Models, Environ. Model. Softw.,
105, 187–200, doi:10.1016/j.envsoft.2018.04.002, 2018a.
- Gidden, M. J., Riahi, K., Smith, S. J., Fujimori, S., Luderer, G., Kriegler, E., Van Vuuren, D. P., Van Den Berg, M., Feng,
L., Klein, D., Calvin, K., Doelman, J. C., Frank, S., Fricko, O., Harmsen, M., Hasegawa, T., Havlik, P., Hilaire, J., Hoesly,
R., Horing, J., Popp, A., Stehfest, E. and Takahashi, K.: Global emissions pathways under different socioeconomic scenarios
1110 for use in CMIP6: a dataset of harmonized emissions trajectories through the end of the century, Geosci. Model Dev.,
doi:10.5194/gmd-2018-266, 2018b.
- Gütschow, J., Jeffery, M. L., Schaeffer, M. and Hare, B.: Extending Near-Term Emissions Scenarios to Assess Warming
Implications of Paris Agreement NDCs, Earth's Futur., 6(9), 1242–1259, doi:10.1002/2017EF000781, 2018.
- Huppmann, D., Rogelj, J., Kriegler, E., Krey, V. and Riahi, K.: A new scenario resource for integrated 1.5 °C research, Nat.
1115 Clim. Chang., 8(12), 1027–1030, doi:10.1038/s41558-018-0317-4, 2018.
- Huppmann, D., Kriegler, E., Krey, V., Riahi, K., Rogelj, J., Calvin, K., Humpenoeder, F., Popp, A., Rose, S. K., Weyant, J.,
Bauer, N., Bertram, C., Bosetti, V., Doelman, J., Drouet, L., Emmerling, J., Frank, S., Fujimori, S., Gernaat, D., Grubler, A.,
Guivarch, C., Haigh, M., Holz, C., Iyer, G., Kato, E., Keramidas, K., Kitous, A., Leblanc, F., Liu, J.-Y., Löffler, K., Luderer,
G., Marcucci, A., McCollum, D., Mima, S., Sands, R. D., Sano, F., Streffer, J., Tsutsui, J., Vuuren, D. Van, Vrontisi, Z.,
1120 Wise, M. and Zhang, R.: IAMC 1.5°C Scenario Explorer and Data hosted by IIASA, Integr. Assess. Model. Consort. Int.

- Inst. Appl. Syst. Anal., doi:10.5281/zenodo.3363345, 2019.
- Lamboll, R. D.: Silicone examples github, , doi:10.5281/zenodo.3987940, 2020.
- Lamboll, R. D., Nicholls, Z. and Kikstra, J.: Silicone documentation, [online] Available from: <https://silicone.readthedocs.io/en/latest/index.html> (Accessed 11 May 2020a), 2020.
- 1125 Lamboll, R. D., Nicholls, Z. and Kikstra, J.: Silicone github, , doi:10.5281/zenodo.3822259, 2020b.
- McKinney, W.: pandas: a Foundational Python Library for Data Analysis and Statistics, Python High Perform. Sci. Comput., 2011.
- Meinshausen, M. and Dooley, K.: Mitigation scenarios for non-energy GHG, in Achieving the Paris Climate Agreement Goals: Global and Regional 100% Renewable Energy Scenarios with Non-Energy GHG Pathways for +1.5C and +2C, pp.
- 1130 79–91, Springer International Publishing., 2019.
- Meinshausen, M., Hare, B., Wigley, T. M. L., Van Vuuren, D., Den Elzen, M. G. J. and Swart, R.: Multi-gas emissions pathways to meet climate targets, *Clim. Change*, 75(1–2), 151–194, doi:10.1007/s10584-005-9013-2, 2006.
- Meinshausen, M., Raper, S. C. B. and Wigley, T. M. L.: Emulating coupled atmosphere-ocean and carbon cycle models with a simpler model, *MAGICC6-Part 1: Model description and calibration*, *Atmos. Chem. Phys.*, 11, 1417–1456,
- 1135 doi:10.5194/acp-11-1417-2011, 2011.
- Meinshausen, M., Vogel, E., Nauels, A., Lorbacher, K., Meinshausen, N., Etheridge, D. M., Fraser, P. J., Montzka, S. A., Rayner, P. J., Trudinger, C. M., Krummel, P. B., Beyerle, U., Canadell, J. G., Daniel, J. S., Enting, I. G., Law, R. M., Lunder, C. R., O’Doherty, S., Prinn, R. G., Reimann, S., Rubino, M., Velders, G. J. M., Vollmer, M. K., Wang, R. H. J. and Weiss, R.: Historical greenhouse gas concentrations for climate modelling (CMIP6), *Geosci. Model Dev.*, 10(5), 2057–2116,
- 1140 doi:10.5194/gmd-10-2057-2017, 2017.
- Nabel, J. E. M. S., Rogelj, J., Chen, C. M., Markmann, K., Gutzmann, D. J. H. and Meinshausen, M.: Decision support for international climate policy - The PRIMAP emission module, *Environ. Model. Softw.*, doi:10.1016/j.envsoft.2011.08.004, 2011.
- Nicholls, Z., Gieseke, R., Lewis, J. and Willner, S.: OpenSCM: Unified access to simple climate models, [online] Available from: <https://github.com/openscm/openscm> (Accessed 6 April 2020), 2020.
- 1145 O’Neill, B. C., Tebaldi, C., van Vuuren, D. P., Eyring, V., Friedlingstein, P., Hurtt, G., Knutti, R., Kriegler, E., Lamarque, J.-F., Lowe, J., Meehl, G. A., Moss, R., Riahi, K. and Sanderson, B. M.: The Scenario Model Intercomparison Project (ScenarioMIP) for CMIP6, *Geosci. Model Dev.*, 9(9), 3461–3482, doi:10.5194/gmd-9-3461-2016, 2016.
- Riahi, K., Rao, S., Krey, V., Cho, C., Chirkov, V., Fischer, G., Kindermann, G., Nakicenovic, N. and Rafaj, P.: RCP 8.5-A
- 1150 scenario of comparatively high greenhouse gas emissions, *Clim. Change*, doi:10.1007/s10584-011-0149-y, 2011.
- Riahi, K., van Vuuren, D. P., Kriegler, E., Edmonds, J., O’Neill, B. C., Fujimori, S., Bauer, N., Calvin, K., Dellink, R., Fricko, O., Lutz, W., Popp, A., Cuarema, J. C., KC, S., Leimbach, M., Jiang, L., Kram, T., Rao, S., Emmerling, J., Ebi, K.,

- Hasegawa, T., Havlik, P., Humpenöder, F., Da Silva, L. A., Smith, S., Stehfest, E., Bosetti, V., Eom, J., Gernaat, D., Masui, T., Rogelj, J., Strefler, J., Drouet, L., Krey, V., Luderer, G., Harmsen, M., Takahashi, K., Baumstark, L., Doelman, J. C., Kainuma, M., Klimont, Z., Marangoni, G., Lotze-Campen, H., Obersteiner, M., Tabeau, A. and Tavoni, M.: The Shared Socioeconomic Pathways and their energy, land use, and greenhouse gas emissions implications: An overview, *Glob. Environ. Chang.*, 42, 153–168, doi:10.1016/j.gloenvcha.2016.05.009, 2017.
- Robiou du Pont, Y. and Meinshausen, M.: Warming assessment of the bottom-up Paris Agreement emissions pledges, *Nat. Commun.*, 9(1), 1–10, doi:10.1038/s41467-018-07223-9, 2018.
- Rogelj, J., Rao, S., McCollum, D. L., Pachauri, S., Klimont, Z., Krey, V. and Riahi, K.: Air-pollution emission ranges consistent with the representative concentration pathways, *Nat. Clim. Chang.*, doi:10.1038/nclimate2178, 2014.
- Rogelj, J., Shindell, D., Jiang, K., Fifita, S., Forster, P., Ginzburg, V., Handa, C., Kheshgi, H., Kobayashi, S., Kriegler, E., Mundaca, L., Séférian, R. and Vilariño, M. V.: Mitigation pathways compatible with 1.5°C in the context of sustainable development, edited by G. Flato, J. Fuglestedt, R. Mrabet, and R. Schaeffer, *Glob. Warm. 1.5 °C an IPCC Spec. Rep. impacts Glob. Warm. 1.5 °C above pre-industrial levels Relat. Glob. Greenh. gas Emiss. pathways*, Context Strength. Glob. response to Threat Clim. Chang., 93–174 [online] Available from: <http://www.ipcc.ch/report/sr15/>, 2018a.
- Rogelj, J., Popp, A., Calvin, K. V., Luderer, G., Emmerling, J., Gernaat, D., Fujimori, S., Strefler, J., Hasegawa, T., Marangoni, G., Krey, V., Kriegler, E., Riahi, K., Van Vuuren, D. P., Doelman, J., Drouet, L., Edmonds, J., Fricko, O., Harmsen, M., Havlík, P., Humpenöder, F., Stehfest, E. and Tavoni, M.: Scenarios towards limiting global mean temperature increase below 1.5 °C, *Nat. Clim. Chang.*, 8(4), 325–332, doi:10.1038/s41558-018-0091-3, 2018b.
- Sanderson, B. M., O'Neill, B. C. and Tebaldi, C.: What would it take to achieve the Paris temperature targets?, *Geophys. Res. Lett.*, 43(13), 7133–7142, doi:10.1002/2016GL069563, 2016.
- Schaeffer, M., Gohar, L., Kriegler, E., Lowe, J., Riahi, K. and van Vuuren, D.: Mid- and long-term climate projections for fragmented and delayed-action scenarios, *Technol. Forecast. Soc. Change*, 90(PA), 257–268, doi:10.1016/j.techfore.2013.09.013, 2015.
- Smith, C. J., Forster, P. M., Allen, M., Leach, N., Millar, R. J., Passerello, G. A. and Regayre, L. A.: FAIR v1.3: A simple emissions-based impulse response and carbon cycle model, *Geosci. Model Dev.*, 11(6), 2273–2297, doi:10.5194/gmd-11-2273-2018, 2018.
- Teske, S., Pregger, T., Simon, S., Naegler, T., Pagenkopf, J., van den Adel, B., Meinshausen, M., Dooley, K., Briggs, C., Dominish, E., Giurco, D., Florin, N., Morris, T. and Nagrath, K.: Methodology, in *Achieving the Paris Climate Agreement Goals*, pp. 25–78, Springer International Publishing, Cham., 2019.
- van Vuuren, D. P., Edmonds, J., Kainuma, M., Riahi, K., Thomson, A., Hibbard, K., Hurtt, G. C., Kram, T., Krey, V., Lamarque, J. F., Masui, T., Meinshausen, M., Nakicenovic, N., Smith, S. J. and Rose, S. K.: The representative concentration pathways: An overview, *Clim. Change*, 109(1), 5–31, doi:10.1007/s10584-011-0148-z, 2011.

1185 Weyant, J.: Some contributions of integrated assessment models of global climate change, Rev. Environ. Econ. Policy,
11(1), 115–137, doi:10.1093/reep/rew018, 2017.

Formatted: Indent: Left: 0 cm, Hanging: 0.5 cm

Author contributions

1190 JR initiated the research based on earlier work by MM. RDL led the code development and the mathematical translation of
infiller methods. RDL and ZRJN wrote the code, JSK assisted reviewing it. JR and MM conceived infilling techniques and
use cases. RDL wrote the manuscript, all authors gave comments and contributed to the final version.

Code availability

The Silicone code in this paper is available from the main github repository (Lamboll et al., 2020b). Code used to analyse the
output of Silicone is available in a second github repository (Lamboll 2020).

1195 Acknowledgements

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant
agreement No 820829 (CONSTRAIN). We thank Nikolai Meinshausen for useful statistics discussions.

1200 Table 1: A guide to crunchers. Names followed by asterisks use a ratio-based approach, i.e. they find a multiplicative factor and
then multiply the infilltarget lead by this value. These crunchers do not preserve harmonisation. If the asterisk is in brackets, a
ratio-based approach is optional. Otherwise, techniques all return linear combinations of values seen in the infiller database.

Name	Description	Use case	Pitfalls
Constant ratio*	Multiplies the lead variable by a constant (not fitted to any data)	Used when no information about the follower variable is available in any database. Mainly used for infilling with zeros.	Has no basis in the data – only used as a last resort in cases of complete uncertainty.
Latest time ratio*	Multiplies the lead variable by a constant fitted to a single (latest) timepoint in the infiller data.	Used when no data is available for most times, this generalises from the latest information we have, e.g. if only historic data is available.	No reason to assume that the relationship between emissions holds for all time. No restriction on signs of follower gas, so potential sign errors when the lead (but not follower) emissions may become negative. Sensitive to emissions trajectories with a high coefficient of variation.

			Emissions may need re-harmonising.
Time dependent ratio*	Multiplies the lead variable by the ratio of the averages of the lead and follower data in the infiller database. (Note: this ratio is not the same as the average of the ratios and is more stable to inclusion of extreme ratios.) Optionally calculates this using only values with the same sign of lead gas -emissions.	Used when two emissions should track each other, or one represents a portion of the other.	Allows arbitrarily high emissions. Can behave unexpectedly if emissions change sign, and an error is produced particularly —if emissions with this sign are not seen at the same time in the infiller database.
RMS closest	Finds the most similar pathway in the infiller database and uses those values. Most similar means smallest root mean squared difference between the lead values of infiller pathways and infilleetarget pathway averaged over all times.	Used when behaviour at one time should strongly determine behaviour at another and continuity is needed between times. The only cruncher that does not treat each time separately.	A small change in the input target data at a single timestep can result in large changes in output at every timestep. All the results returned are found exactly in the infiller database, so if that database is small, the same values are returned in many cases. Results more extreme than found in the infiller database all return the same value.
Linear interpolation / Interpolate specified scenarios and models	At each time, linearly interpolates between the follower values at the two nearest lead values, taking averages where multiple points have identical lead values. Interpolate specified scenarios and models filters the input-infiller database before applying the same technique.	Used for infilling where we have a small number of comparable models/scenarios. The required filtering gives control over the narrative used for infilling.	A small change in the input target data can result in a large change in the output at the same timestep because pathways in the infiller database can be very different in follower variables for nearly identical values of the lead variable. For similar reasons, results can vary erratically between timesteps for large infiller datasets. Results more extreme than those found in the infiller database all return the same value.
Quantile rolling window s (QRW)/time-dependent quantile rolling windows s (*)	At each time, applies a $1/(1+(\text{lead variable difference})^2)$ weighting to datapoints at equally spaced points across the infiller lead. Then calculates a specified quantile (usually the median) for the infiller follower value	Can choose options to give more smoothing (less noise) or more localised behaviour (shows trends better) Allows the option to generate a distribution of outputs, not just a single optimum. Can add to the	Using with any quantile larger than 0.5 will result in all emissions being higher, even if the lead and follower emissions anticorrelate. Results more extreme than found in the infiller database all return the same value.

	at these points. Can also be used in ratio mode, in which case the ratio between lead and follower in the infiller database is treated as above. Time-dependent QRW allows the quantile to be different at different times (but is computationally slower).	narrative through time-dependence. Ratio mode allows better infilling outside the range of the input -infiller data.	unless in ratio mode. In ratio mode, sign changes in the lead variable can result in follower emissions being assigned undesired negative values.
Equal quantile walk (EQW)	Calculates the quantile of the infiller database corresponding to the lead value in each individual year. Returns that quantile in that year of the follow value from the same database.	Conceptually simple, used by previous work.	Assumes all variables are monotonically increasing together. Results more extreme than those found in the infiller database all return the same value.

Table 2: Guide to aggregation tools and multiple infillers. Names followed by asterisks use a ratio-based approach, i.e. they find a multiplicative factor and then multiply the target lead by this value, if the asterisk is in brackets there are ratio-a.

Name	Description	Use case	Pitfalls
Aggregation tools			
Aggregate to composite values	Requires only the infiller target database. Adds together known values to construct a consistent output (with optional weighting).	Infilling aggregate values (e.g. Kyoto gas totals) or finding remainders given aggregates and values for the other components.	Requires all information to be known already – no statistical inference, just adding.
Multiple infillers			
Decompose collection with time-dependent ratio *	Constructs a consistent version of the aggregate in the infiller database. Breaks a known quantity down into components, estimated by the time-dependent ratio method.	Breaking down aggregate values into their components, assuming all should be treated similarly.	Requires all infiller scenarios which do not have to have values for all components at all times are ignored. Ignores the aggregate if the infiller database has inconsistency between that and the sum of reported components. Assumes direct proportionality between components and sum, which is problematic around sign changes. otherwise ratios will be distorted. Any contributions to the aggregate values not quantified in the infiller database are ignored.
Split collection with remainder	Breaks an aggregate emission into most of its separate components, with	Breaking down aggregate values into their component when one	The remainder emission is not constrained, nor as precisely estimated as the other values.

Formatted Table

Formatted Table

Formatted Table

<u>emissions</u>		<u>one emission type making up the remainder of the emissions.</u>	<u>emission type is much larger than the others, or may be either positive or negative</u>	
Infill required values (*)	all	Uses the same lead variable and cruncher to infill any gaps in emissions data.	For infilling scattered, minor gaps in a largely sound database.	Low confidence in the results being accurate as the method does not consider the specific characteristics of the data.

Table 3. Absolute values of Spearman's Rank correlation between emissions, averaged over the start of decades from 2020 to 2100. We use the following abbreviations: BC as black carbon, VOC as volatile organic compounds, AFOLU as Agriculture, Forestry and Other Land Use; and En & IP as energy and industrial processes. "CO₂" represents subtypes of CO₂. We also calculate the average of these rows, with or without the CO₂ and subtypes. Cells are bold and yellow if the value in them is > 0.7 and are bluecoloured-orange if the variance of the rank correlation between years exceeds 0.03. There is no overlap between these categories.

Formatted: Su

variable	BC	CH ₄	CO	CO ₂	CO ₂ + AFOLU	CO ₂ +En-& IP	F-gases	HFC	N ₂ O	NH ₃	NO _x	OC	PFC	SF ₆	Sulfur	VOC
BC		0.47	0.75	0.46	0.37	0.42	0.23	0.10	0.40	0.40	0.58	0.73	0.41	0.20	0.48	0.45
CH ₄			0.32	0.74	0.49	0.73	0.64	0.58	0.86	0.34	0.58	0.30	0.66	0.41	0.65	0.24
CO				0.36	0.38	0.32	0.06	0.16	0.29	0.35	0.48	0.78	0.05	0.17	0.36	0.68
CO ₂					0.54	0.96	0.60	0.57	0.54	0.30	0.61	0.24	0.35	0.22	0.69	0.37
CO ₂ +AFOLU						0.36	0.27	0.40	0.53	0.36	0.33	0.34	0.23	0.21	0.31	0.20
CO ₂ +Energy & IP							0.58	0.51	0.50	0.25	0.61	0.17	0.32	0.18	0.69	0.36
F-gases								0.19	0.57	0.19	0.50	0.10	0.90	0.77	0.60	0.12
HFC									0.46	0.11	0.30	0.14	0.71	0.68	0.36	0.23
N ₂ O										0.44	0.46	0.30	0.65	0.40	0.49	0.17
NH ₃											0.23	0.39	0.10	0.05	0.23	0.25
NO _x												0.22	0.53	0.26	0.76	0.30

[illegible]

<u>SF6</u>															0.26	0.24
<u>Sulfur</u>															-	0.46
<u>VOC</u>																
<u>Average</u>	0.43	0.53	0.37	0.50	0.36	0.46	0.47	0.42	0.47	0.27	0.46	0.31	0.43	0.33	0.47	0.32
<u>Average,</u> <u>no CO₂</u>	0.43	0.50	0.37	0.46	0.34	0.43	0.47	0.40	0.46	0.26	0.44	0.32	0.47	0.36	0.44	0.32
<u>#</u> <u>scenario</u> <u>s</u>	389	412	353	414	412	414	368	108	411	345	363	363	180	191	412	345

Formatted: Fo

Formatted: Fo

Formatted: Fo

Formatted: Fo

Formatted Ta

Formatted: Fo

Formatted: Fo

Formatted: Fo

Formatted: Fo
pt, Subscript

Formatted: Fo

Formatted: Fo

Formatted: Fo

1215

[illegible]

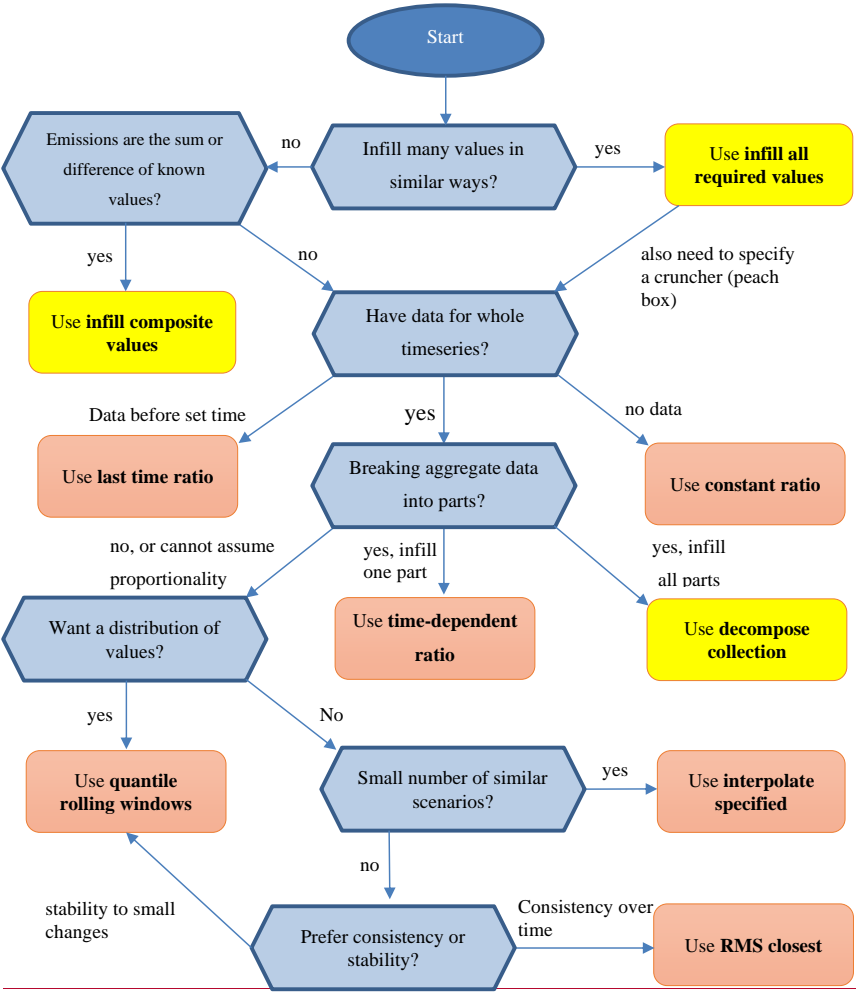
1220

41

Mean	0.9130.128	0.6450.102	0.5210.152	0.7710.155	0.778
------	------------	------------	------------	------------	-------

- Formatted: Font: (Default) +Body (Times New Roman), 10 pt, Bold
- Formatted: Font: (Default) +Body (Times New Roman)
- Formatted: Font: (Default) +Body (Times New Roman), 10 pt, Bold
- Formatted: Font: (Default) +Body (Times New Roman)
- Formatted: Font: (Default) +Body (Times New Roman), 10 pt, Bold
- Formatted: Font: (Default) +Body (Times New Roman)
- Formatted: Font: (Default) +Body (Times New Roman), 10 pt, Bold
- Formatted: Font: (Default) +Body (Times New Roman)
- Formatted: Font: (Default) +Body (Times New Roman), 10 pt, Bold
- Formatted: Font: (Default) +Body (Times New Roman)

Figure 1: Flow chart suggesting how to choose the cruncher (peach oblongs) or multiple infiller (yellow oblongs) to use when infilling.



Field Code Changed

Formatted: Font: Not Bold

Formatted: Font: 9 pt, Bold

Formatted: Caption

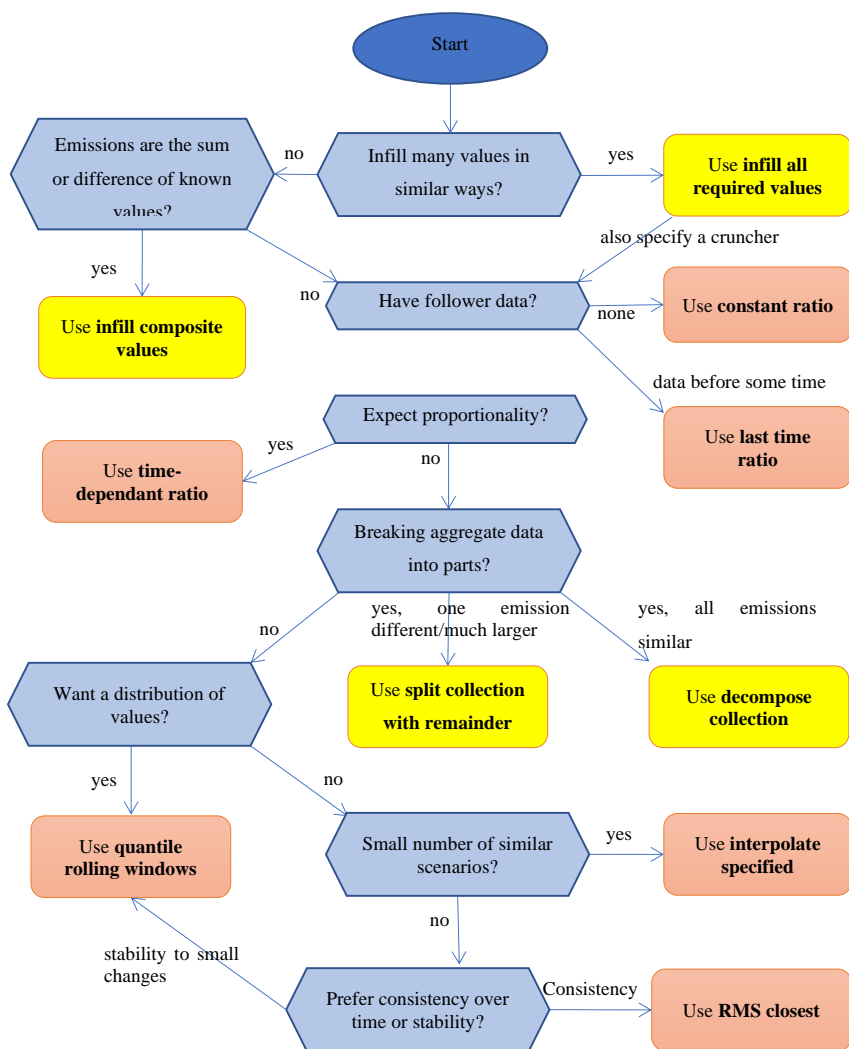
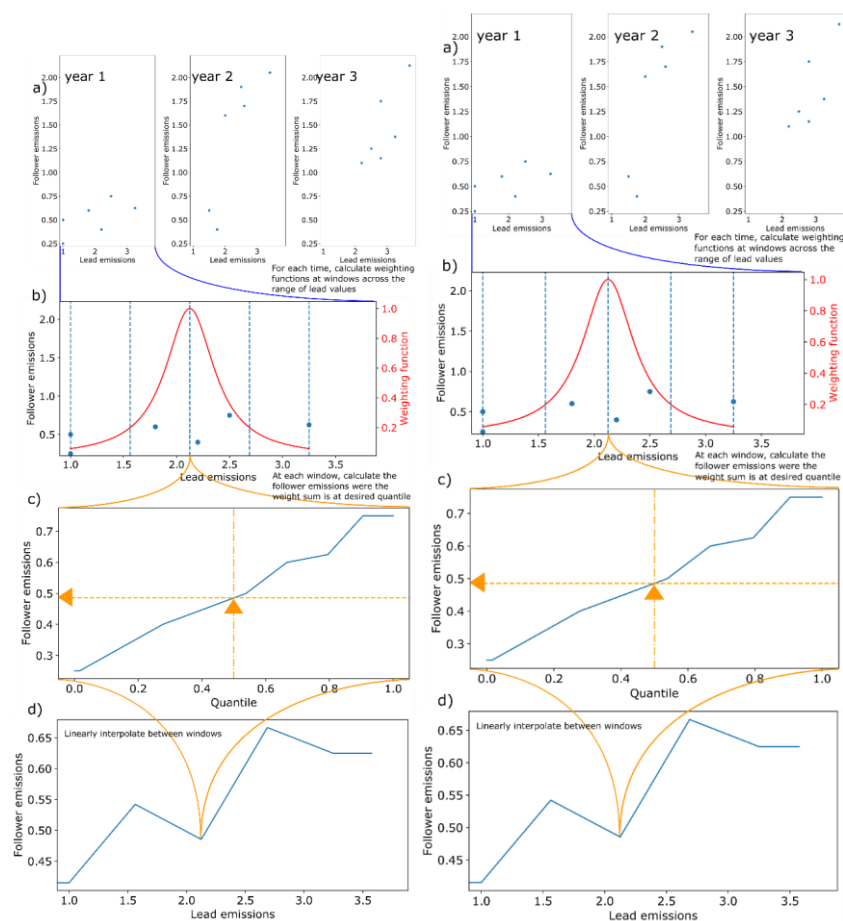


Figure 2: Schematic of how the quantile rolling windows cruncher determines the follow value to use. a) Example relationships between lead (CO₂) and follow (CH₄) variables over time. b) A number of rolling windows (here 5, by default 10) are drawn and a weighting function constructed for each window. It has a continuous distribution, rather than a discrete cutoff, hence the name. c) A relationship between the sum of the weights and the follow value is established and the follow value at the desired quantile is returned.



Formatted: Normal

1235

| Figure 3: Left: The ~~Shell-Sky-model-POEM scenario B projection for~~ CO₂ from Energy and Industrial Applications data. The fine
| lines represent the different timeseries in the SR1.5 database used to perform the infilling and are not included in the legend for
| clarity. Right: The results of interpolating this data using ~~four-five~~ different crunchers. The interpolate specified model approach
1240 | used ~~the MESSAGE model and only choses scenarios based on SSP2 pathways,the REMIND-MAGPIE 1.5 model and only choses~~
| ~~scenarios based on SSP5 pathways.~~

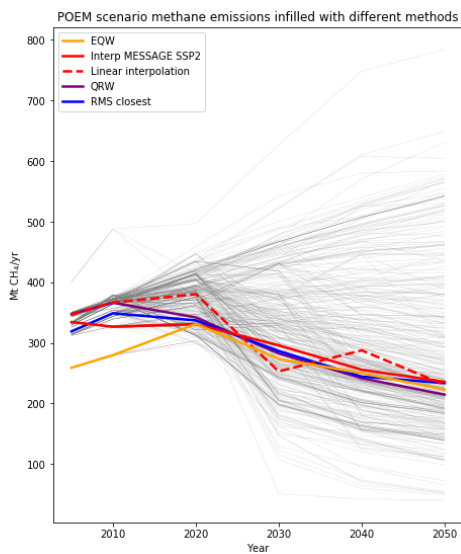
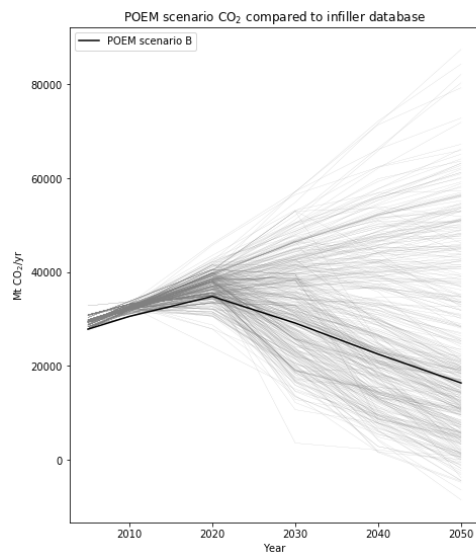
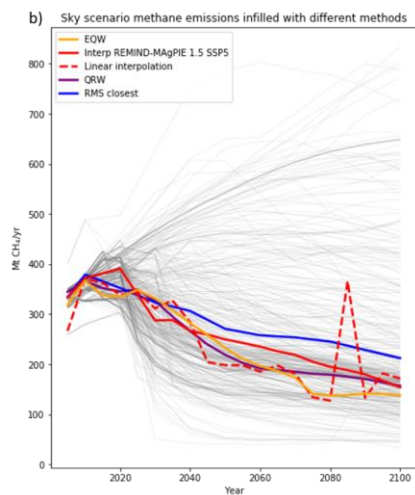
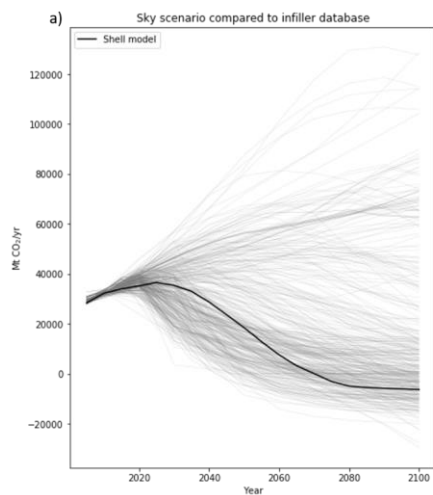


Figure 4: The Climate Action Tracker (CAT) Kyoto gas totals (thick lines) compared with the portfolio of values in the SR1.5 database (thin lines).

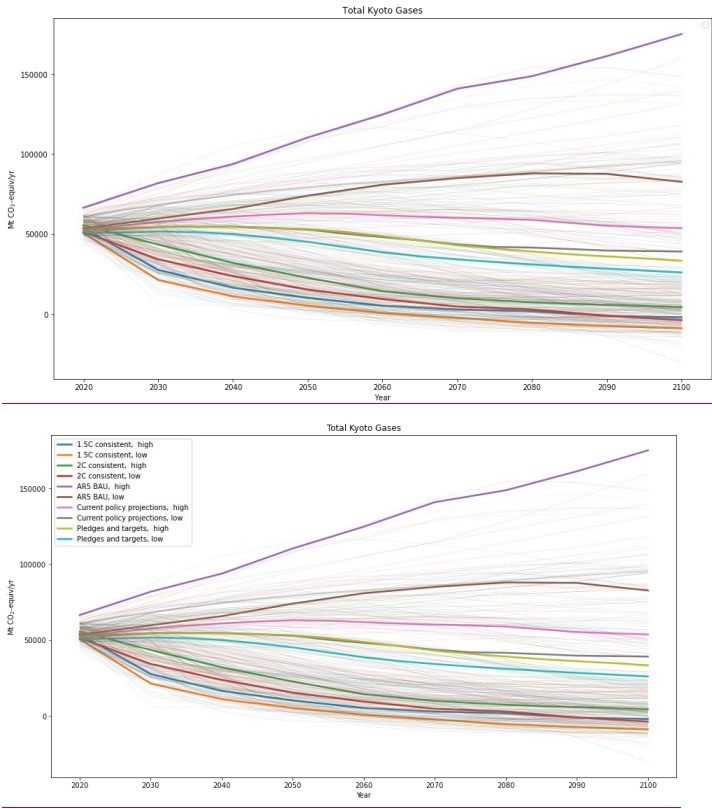


Figure 5: The CAT Kyoto gas baskets decomposed into their components, using the decompose collection multiple infiller.

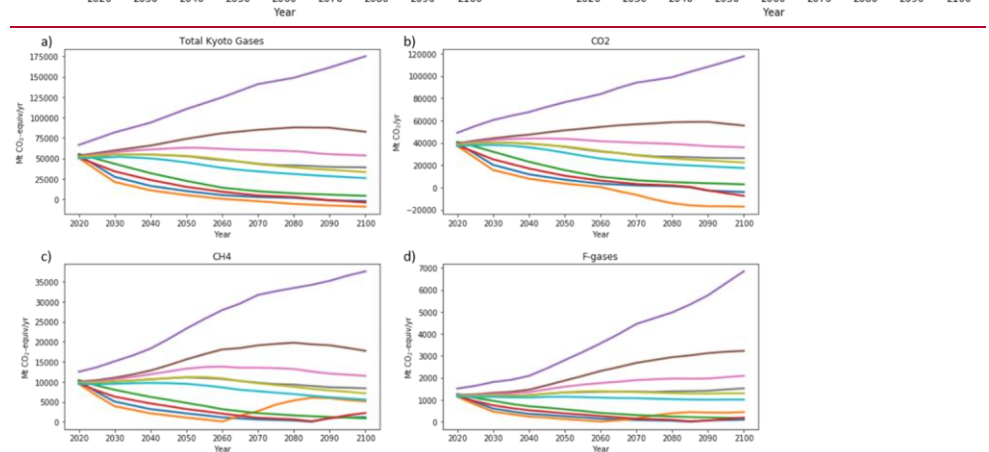
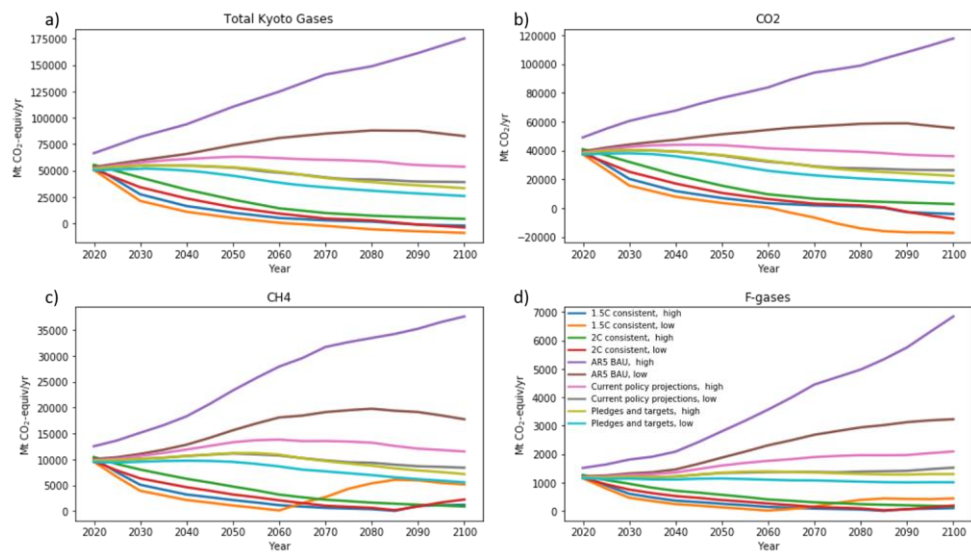


Figure 6: Kyoto gases, decomposed by first infilling the non-negative emissions using the (non-ratio) quantile rolling windows, then infilling the CO₂ using infill composite values.

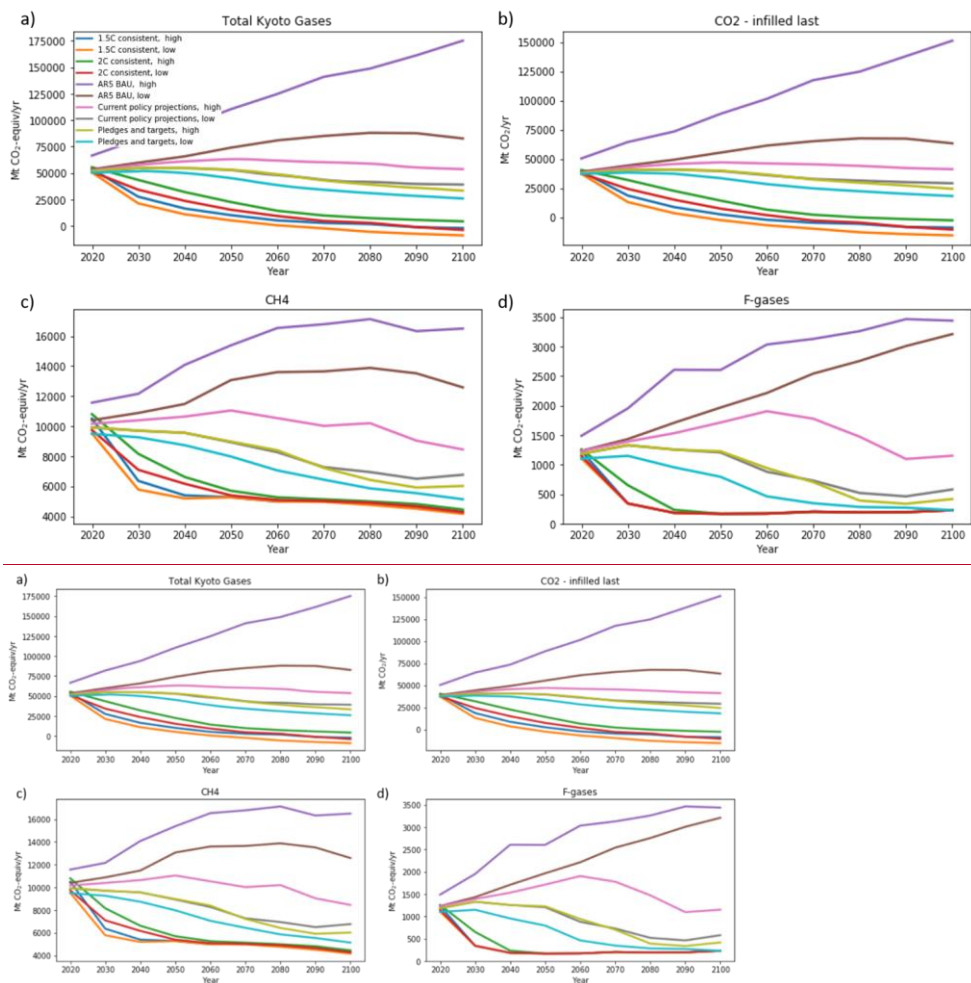
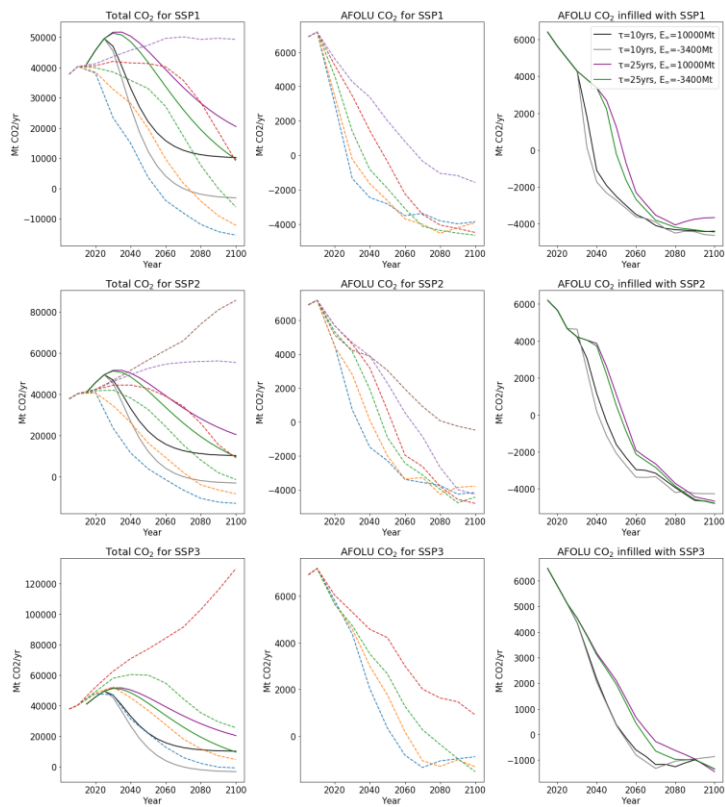


Figure 7: Illustration of using the interpolate specified scenario cruncher to infill a series of stylised trajectories (solid lines), characterised by two different parameters (τ and E_{∞}), defined in (Sanderson et al., 2016). The first column compares the total CO₂

1260 calculated for the stylised trajectories to the values of the MESSAGE model for a given group of SSP scenarios (dotted lines). These are our lead values in each case. The second column shows the range of follow values for that SSP. The third column shows the resultant AFOLU (Agriculture, Forestry and Other Land Use) trajectories that emerge from using the Interpolate Specified Scenario infiller.



1265