Table 2. Further details on the explanatory variables used (sorted by the physical scheme and variable name)

Explanatory variable*	Variation variable[†]	Physical scheme	Basic meaning	What the parameter does physically in the model or other explanation
eacf	–	Cloud physics	Empirically adjusted cloud fraction	This calculates how much cloud cover there will be when the air is saturated.
rhcrit	–	Cloud physics	Critical relative humidity	This parameter accounts for the fact that when clouds form they do not have to fill a whole grid box. The whole grid box does not have to be at saturation for clouds to form—instead clouds are allowed to form at a lower threshold critical relative humidity. This threshold is calculated based on a distribution function of cloud sizes.
ct	–	Cloud physics and precipitation	Accretion constant	This relates how quickly cloud droplets convert to rain.
cw	–	Cloud physics and precipitation	Threshold for precipitation	This relates how much water there is in a cloud to when it starts raining, which is dependent on the condensation nuclei concentration—the more condensation nuclei there are (bits of dust, salt, etc., in the atmosphere on which raindrops can form) the smaller the raindrops.
vf1	–	Cloud physics and precipitation	Ice fall speed through clouds	This is important for the development of clouds and determining type (rain, sleet, hail, snow) and amount of precipitation
ice	–	Cloud physics and shortwave radiation	Nonspherical ice	Allows for nonspherical ice particles in the radiation scheme.
ice_size	–	Cloud physics and shortwave radiation	Ice particle size	This gives an effective radius for ice crystals in clouds—i.e., what radius would they have if they were perfectly spherical. It is important in the radiation scheme, to calculate how much incoming or outgoing radiation is reflected, etc.
entcoef	–	Convective cloud physics	Entrainment coefficient	This parameter determines how rapidly a convective cloud mixes in clear air from around it.
dtheta		Initial conditions	Perturbations to initial conditions on a given level	Small changes in the initial temperature results in models developing different "weather"—as in Lorenz's infamous butterfly effect.
alpham	–	Sea ice and short wave radiation	Albedo at melting point of ice	Sets the albedo of sea ice at ice melting point. This parameter along with dtice accounts for temperature variations in the reflectivity of ice.
dtice	–	Sea ice and shortwave radiation	Temperature range of ice albedo variation	Sets the range of temperatures over which the reflectivity of sea ice varies.

Explanatory variable*	Variation variable[†]	Physical scheme	Basic meaning	What the parameter does physically in the model or other explanation
clock_boinc_f	clock_boinc_f_CV	Software and hardware	Floating point processor clock speed recorded under BOINC middleware	Prior to running the application, and intermittently throughout the application's run time, the BOINC middleware runs a series of tests to assess the speed of the client computer it is running on. These tests are designed to be independent of CPU architecture, number of megahertz, revision, etc. This variable stores the result of a test to determine the number of floating point operations, that is, arithmetic operations involving numbers with fractional parts, a client computer can complete within a set period of time.
clock_boinc_i	clock_boinc_i_CV	Software and hardware	Integer processor clock speed recorded under BOINC middleware	Similarly to clock_boinc_f, this variable records the result of a test to determine the number of integer operations, that is, arithmetic and logical operations involving whole numbers, a client computer can complete within the same set period of time. It is important to distinguish between clock_boinc_f and clock_boinc_i as floating point and integer operations are handled by different units within the CPU. The application may be floating point or integer operation intensive, and so, assessing the speed of the CPU in such a manner allows the BOINC middleware to approximate how long the application will take to run to completion.
clock_classic	clock_classic_CV	Software and hardware	Processor clock speed recorded under classic middleware	This records the raw speed, in kilohertz, of the client computer. This is the only metric of CPU speed available under the classic middleware (see below). It is important to distinguish this variable from those measuring the client speed under the BOINC middleware as this variable is not a direct measure of the CPU speed, rather it is an indirect measure of how well the CPU is expected to perform, within its class. CPUs of the same architecture will perform better as the raw speed increases whereas two CPUs of the same number of kilohertz but with different architectures may perform entirely differently.

Explanatory variable*	Variation variable[†]	Physical scheme	Basic meaning	What the parameter does physically in the model or other explanation
midware	–	Software and hardware	Client middleware	Middleware is software that implements routines common to a class of applications. These routines are built into a library that the application can be built upon or alongside. In distributed computing, the middleware is the underlying network code that facilitates communication to the project servers and the mechanisms by which client machines join the project, obtain tasks, and submit their results. There are two parts to the network middleware, the client part, running on a participant's computer and the server part, running on the project's servers. The climate*prediction*.net program running on a participant's computer consists of two distinct components, the climate modeling application and the application built upon the middleware, the network client application. These two applications run concurrently and share information such as the percentage of work complete. There also is a visualization component to the client application that allows the status of the climate modeling application to be represented graphically. Prior to August 2003, climate*prediction*.net used a proprietary middleware ("classic") to perform the network communication between the participant's computer and the project's servers. This was developed by the climate*prediction*.net team. In August 2003, climate*prediction*.net switched to using the Berkeley Open Infrastructure for Network Computing (BOINC) middleware. This was developed by the team at University Of California, Berkeley, responsible for the SETI@home project (3). The BOINC middleware is now in use by over 20 different projects, all sharing the same underlying networking and scheduling code.

Explanatory variable*	Variation variable[†]	Physical scheme	Basic meaning	What the parameter does physically in the model or other explanation
os_name	–	Software and hardware	Operating system	Initially, climate*prediction*.net was available for the Microsoft Windows family of operating systems, that is Windows 98, ME, 2000, and XP. After the move to the BOINC middleware, climate*prediction*.net was made available for the aforementioned Windows and also the Linux and Apple Macintosh OS X operating systems. When the client connects to the project server for the first time, it communicates the type of operating system the client is running, which then is stored in a database on the server with other information about the client and the task it currently is performing. The type of operating system is retrieved from an operating system call as a text string. This text string is not always a complete representation of the operating system and so some processing was carried out to classify the operating systems into distinct classes. Over 600 different operating systems present in the database were reclassified into 13 operating systems: Windows 95, 98, ME, XP, 2000, 2003, NT4; Mac OS X; Linux, OpenBSD, NetBSD FreeBSD and Unknown, however, none of the BSD operating systems was found in the dataset. It is important to note which operating system the climate modeling application is running under as the variation in implementation of parts of the operating system, especially the implementation of mathematical routines and time measuring routines, could have a large bearing on the outcome of the computation.
processor_name	–	Software and hardware	CPU classification	Similarly to the os_name, the processor name is returned when the client connects to the server for the first time. The name of the processor is returned by an operating system call, and again it is imperfect information. Since using the BOINC middleware, climate*prediction*.net is available to run on PowerPC and x86 processors. The processors were reclassified into: x86, PowerPC and unknown. x86 was subdivided into Intel, AMD, VIA, Transmeta and unknown. These classifications were further subdivided into most of the processors available in the past ten years. However, the processor type "unknown x86" accounted for 20% of the processor types in the database. It is important to note the processor type the application is running on as differences in their architecture, especially the floating point unit, can have an effect on the result of the computations.
ram_size	RAM_CV	Software and hardware	Hardware RAM	Similarly to the os_name and processor_name, the ram (memory) size of a client is recorded in the database on the server. When an operating system runs out of physical memory, it pages some of the memory to disk. Computers with more memory will page to disk less often than those with less memory. Differences in ram size may influence the computations due to this.

More information about parameters in the HadAM3 model, including those not varied in this study are available in ref. 1 and the supplementary information to ref. 2.

*The name of the explanatory variables used in the trees in Fig. 2, SI Fig. 8, and SI Tables 3 and 4.

†Name of the alternative variable used when the response variable is a coefficient of variation (CV), i.e., in Fig. 6, SI Fig. 10, and SI Tables 5 and 6 (in cases where no variation variable is given, the same variable was used as in the other trees).

1. Barnett DN, Brown SJ, Murphy JM, Sexton DMH, Webb MJ (2006) *Clim Dyn* 26:489-511.

2. Murphy JM, Sexton DM, Barnett DN, Jones GS, Webb MJ, Collins M, Stainforth D A (2004) *Nature* 430:768-772.

3. Anderson DP, Christensen C, Allen B (2006) in *Supercomputing 06-The International Conference for High Performance Computing, Networking, Storage, and Analysis* (Supercomputing, Kearney, MO), pp 33-42.