A connectome manipulation framework for the systematic and reproducible study of structure–function relationships through simulations

Christoph Pokorny[1], Omar Awile[1], James B. Isbister[1], Kerem Kurban[1], Matthias Wolf[1], and Michael W. Reimann[1]

[1]Blue Brain Project, École Polytechnique Fédérale de Lausanne (EPFL), Campus Biotech, 1202 Geneva, Switzerland

Abstract

Synaptic connectivity at the neuronal level is characterized by highly non-random features. Hypotheses about their role can be developed by correlating structural metrics to functional features. But to prove causation, manipulations of connectivity would have to be studied. However, the fine-grained scale at which non-random trends are expressed makes this approach challenging to pursue experimentally. Simulations of neuronal networks provide an alternative route to study arbitrarily complex manipulations in morphologically and biophysically detailed models. Here, we present Connectome-Manipulator, a Python framework for rapid connectome manipulations of large-scale network models in SONATA format. In addition to creating or manipulating the connectome of a model, it provides tools to fit parameters of stochastic connectivity models against existing connectomes. This enables rapid replacement of any existing connectome with equivalent connectomes at different levels of complexity, or transplantation of connectivity features from one connectome to another, for systematic study. We employed the framework in a detailed model of rat somatosensory cortex in two exemplary use cases: transplanting interneuron connectivity trends from electron microscopy data and creating simplified connectomes of excitatory connectivity. We ran a series of network simulations and found diverse shifts in the activity of individual neuron populations causally linked to these manipulations.

Keywords — Connectome; manipulation; rewiring; neural networks; structure–function; SONATA

1 Introduction

The structure of synaptic connectivity decidedly shapes neuronal activity. It can even be said to implement the specific functions of different microcircuits. For example, attractor states have been shown to emerge in models with clusters of neurons that are more strongly interconnected than the rest of the population (Litwin-Kumar and Doiron, 2012; Deco and Hugues, 2012; Lagzi and Rotter, 2015). Studying the link between structure and function becomes harder for more complex trends of connectivity, such as overexpression of triad motifs or targeting specificity (Song et al., 2005; Perin et al., 2011; Schneider-Mizell et al., 2023; Pi et al., 2013). Yet, increased complexity that is not captured by comparatively simple connectivity model has been demonstrated to be relevant. For example, clustered inhibition allows competition between attractors without firing rate saturation (Rost et al., 2018), and Renart et al. (2007) speculated that beyond the topology of the wiring diagram, biological details such as dynamic synapses, synaptic failure, dendritic integration, and synaptic clustering may be crucial.

The advent of electron-microscopic (EM) tissue reconstructions, such as the MICrONS mmˆ3 dataset (MICrONS Consortium et al., 2021), has been a great boon for researchers studying such important questions, as they provide a complete snapshot of neuronal connectivity instead of sparsely sampled connections. The research to find and describe the mechanisms of connectivity in these datasets that implement the local circuit's function is still ongoing. However, large-scale EM connectomes contain millions of synapses between thousands of neurons, allowing the discovery of any number of non-random trends. But which ones are functionally relevant, and which ones are mere epiphenomena? Co-registered neuron recordings enable their correlation with function, but to demonstrate causation, a change of function must be the result of a manipulation that affects the structural trend observed. Currently, the only viable way to conduct such an experiment is *in silico*. Recently, modeling techniques have been developed that include sufficient biological detail to reproduce non-random connectivity trends observed in biology (Markram et al., 2015; Billeh et al., 2020; Isbister et al., 2023). An even more powerful approach could be based on *transplanting* the connectivity observed in EM into an *in silico* model. Both cases can then be followed by *connectome manipulations* that add or remove connectivity trends, together with observations of their functional impact.

While promising, the outlined research faces confounding factors and particular difficulties: activity *in silico* is affected by simplifications and assumptions that are inherent to the process of modeling. As such, any *in silico* model should be carefully validated against experimental data, and the effect of connectome manipulations should be probed in different baseline models to assess robustness of the results. Other challenges arise from the fact that detailed, bottom-up models include multi-synaptic connectivity (Reimann et al., 2015), i.e., multiple and individually parameterized synapses forming a connection from one neuron to another. This usually leads to large data sizes and complicated file structures that are necessary to store connectivity at such a level of detail. Hence, manipulations must not only be efficient enough to deal with large amounts of data but also preserve biological distributions of parameter values and respect known biological principles, e.g., Dale's law (Strata et al., 1999). Additionally, non-random trends of interest exist in the higher-order structure of connectivity (Song et al., 2005; Perin et al., 2011) or at the level of subcellular targeting of pathways (Pi et al., 2013; Schneider-Mizell et al., 2023). Therefore, specifically manipulating them while keeping the connectivity otherwise unchanged can be conceptually or mathematically challenging.

Here, we present Connectome-Manipulator, a programmatic Python framework that enables the creation, transplantation, and manipulation of connectomes. Based on the open SONATA standard (Dai et al., 2020), the framework allows manipulations to be applied to any network model described in that standard, while their functional impact can be readily investigated through network simulations using any simulator supporting SONATA. Notably, SONATA is based on a non-stochastic representation of a network that explicitly includes multi-synaptic connections by storing each individual synapse and its parameters together with its pre- and post-synaptic neuron. This allows the representation of network models at any level of topological complexity as well as connectome reconstructions from EM. On the one hand, we offer as part of the framework algorithms to systematically simplify the connectivity of a given SONATA model. While simplifying the structure of the wiring diagram, other anatomical and physiological parameters are carefully preserved. This allows the user to start with any proposed complex connectome and then study the impact of simplification. On the other hand, we also provide functionality to add new complexity to a connectome, as defined by an arbitrarily complex adjacency or synaptome matrix. While an adjacency matrix represents the connectivity between pairs of neurons, a synaptome matrix can be used to even define the exact number of synapses that are part of a given connection, thereby offering great flexibility in studying the effect of specific manipulations while preserving other aspects of connectivity.

In addition, while not the main focus, the framework also enables basic manipulations, such as adjustments of physiological synapse parameters, or specific removal of synapses or connections as in lesion experiments. While both SONATA and our framework were developed with morphologically and physiologically detailed models in mind, they also support point neuron models. Thus our framework can be used for example to transplant a wiring diagram from a detailed to a point neuron model, in order to study manipulations on that level.

We demonstrate the applicability of our framework in two typical use cases, by manipulating the connectome of a detailed model of the rat somatosensory cortex (Reimann et al., 2022; Isbister et al., 2023) in two particular ways. First, we increased inhibitory targeting specificity of VIP+ (vasoactive intestinal peptide-expressing) interneurons, thereby transplanting connectivity trends present in the MICrONS dataset (Schneider-Mizell et al., 2023). We found that despite the fact that the VIP+ interneurons are predominantly targeting other inhibitory neurons in the manipulated connectome, their activation can still lower the firing rate of excitatory populations, but to a lesser extent than in the original connectome. Second, we studied how progressively simplified (Gal et al., 2020), but otherwise equivalent connectivity among excitatory neurons affects the dynamics of spiking. We found that layer 4 excitatory neurons were consistently shifted towards a more internally driven spontaneous activity regime, while layer 6 inhibitory neurons were shifted towards a more externally driven regime. For layer 6 excitatory neurons we found diverse effects depending on the degree of simplification. Taken together, these results demonstrate that our framework allows emergent network activity to be causally linked to specific features of connectivity in a systematic and reproducible way.

2 Results

2.1 Connectome manipulation framework

The Connectome-Manipulator software presented in this manuscript is a universal framework for creating and manipulating connectivity in large-scale models of neural networks in SONATA format (Dai et al., 2020). Its source code is openly available on GitHub under `https://github.com/BlueBrain/connectome-manipulator`. Manipulations can be applied to entire models, specific sub-networks, or even single neurons, ranging from insertion or removal of specific motifs to complete rewiring based on stochastic connectivity models at various levels of complexity. Important scientific use cases include *wiring* a connectome from scratch based on given connectivity rules, *rewiring* an existing connectome while preserving certain aspects of connectivity, and *transplanting* specific connectivity characteristics from one connectome to another (Figure 1A).

The *connectome manipulation* functionality (Figure 1B) applies one or a sequence of manipulations to a given connectome of a network model in SONATA format, consisting of a set of nodes (neurons) with or without edges (synapses) between them. Manipulations can be targeted to the entire connectome or to selected pathways, i.e., connections between specific pre- and post-synaptic neurons, based on criteria such as their morphological type (*m-type*) or electrophysiological type. The output of such manipulation(s) is again a network model in SONATA format consisting of the same set of neurons as the original network, together with the manipulated connectome. A typical experimental workflow starts with the creation of a set of manipulated connectomes from a given baseline connectome. The resulting connectomes can be readily simulated using any simulator supporting SONATA, allowing the systematic and reproducible characterization of causal effects of structural manipulations on the network activity and function.

Conceptionally, a biophysically detailed connectome can be described on different levels of detail (Figure 1C and Methods): The *synapse* level preserves the full level of detail, describing individual synapses and their anatomical and physiological parameters. The *connection* level simplifies this to an adjacency matrix, i.e., representing whether or not a connection from one neuron to another exists. The *stochastic model* level simplifies this further to a stochastic description, such as a notion of distance-dependent connection probability. Note that within this level, various types of stochastic model descriptions with different amounts of detail exist (see Table S1). At this level, physiological parameters are also stochastically described by probability distributions. Our framework provides tools for *model building* that fit stochastic model parameters against existing connectomes (see Table S2). This leads to six classes of connectome manipulations that differ in the levels of descriptions they employ, and where manipulations are applied to (Figure 1C):

Alter synapses (red): Keeps the description on the level of synapses and changes physiological parameters, such as conductance.

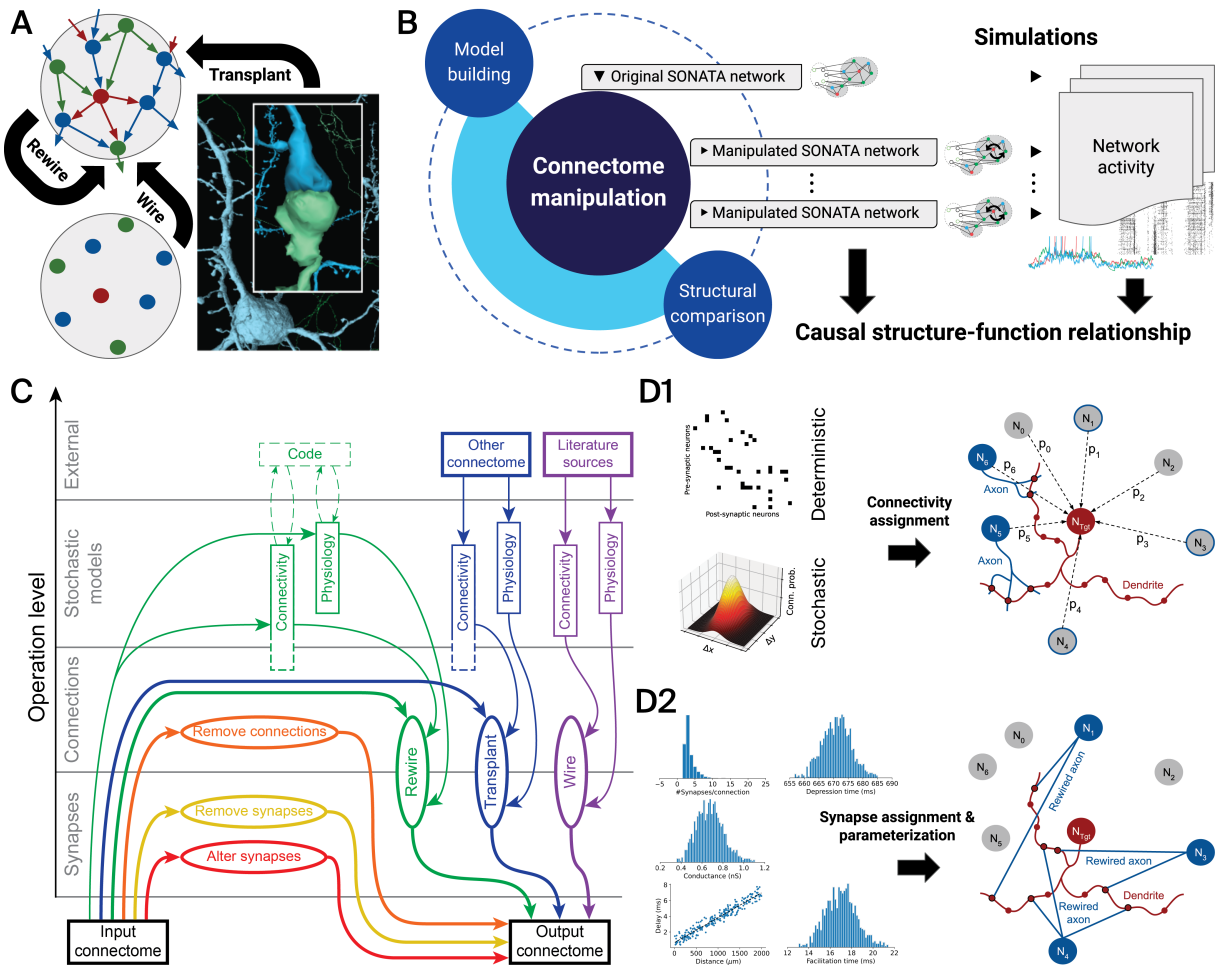

Figure 1: Connectome manipulation framework. A: Scientific use cases, such as wiring or rewiring a connectome based on connectivity rules, or transplanting specific connectivity characteristics from other connectomes (e.g., from EM reconstructions from MICrONS Consortium et al. (2021) as shown here). **B:** Typical experimental workflow, which involves creating a set of manipulated connectomes and running computer simulations, to causally link the emergent network activity to certain structural features of connectivity. **C:** Operation levels and how they interact in different connectome manipulation scenarios, as indicated by color. **D:** Algorithmic steps for establishing new connections: First, connectivity assignment, supporting deterministic and stochastic descriptions of connectivity (D1). Second, synapse assignment and physiological parameterization, allowing pathway-specific parameter distributions (D2).

Remove synapses (yellow): Keeps the description on the level of synapses but removes some of them according to specified selection criteria of pre- and post-synaptic neurons.

Remove connections (orange): Works similarly but on the level of connections, i.e., removing entire connections according to specified selection criteria of pre- and post-synaptic neurons.

Rewire (green): Rewires an existing connectome, that is, creating new synapses based on a description on the level of connections or stochastic models. Existing synapses may be kept or removed. Additional adjustments can optionally be performed manually at this stage or be implemented in code (dashed lines). This can be done independently for the structure and physiology of connections.

Transplant (blue): Same as rewiring, but the source of the connectivity and/or physiological description is another connectome.

Wire (purple): Same as transplant, but the input connectome is empty. The connectivity and physiological description is taken from experimental data or literature.

At the end of a connectome manipulation an output at the level of synapses is generated. For rewire, transplant and wire operations, an instance at the level of connections is built first (Figure 1D1 and Methods). This is then further expanded into a description at the level of synapses by placing synapses, assigning them to connections, and parameterizing them by evaluating the physiological, stochastic model descriptions (Figure 1D2 and Methods). Simplifying a connectome to the level of a stochastic model will usually drastically alter many aspects of connectivity, such as dendritic locations of synapses from different sources, or in-degree distributions. Various options exist in order to carefully preserve such aspects during this process (Table 3).

2.2 Rewiring of VIP+ interneuron connectivity in a detailed model of rat somatosensory cortex

We employed the connectome manipulation framework to rewire interneuron connectivity in a detailed anatomical (Reimann et al., 2022) and physiological (Isbister et al., 2023) network model of the rat somatosensory cortex (Figure 2A and Methods). We introduced a preference for VIP+ interneurons to target other inhibitory neurons with their connections (Pfeffer et al., 2013; Pi et al., 2013) that was not present in the original network model (Reimann et al., 2022). To that end, we separately built a connectivity matrix that is consistent with the targeting preference for VIP+ neurons quantified by Schneider-Mizell et al. (2023) in the MICrONS dataset (MICrONS Consortium et al., 2021) (see Methods). We then used our framework to transplant the matrix into the network model. Physiological parameters of the new connections were drawn from stochastic models fit to the original connectome. Specifically, we rewired pathways originating from VIP+ interneurons, i.e., of m-types bitufted cell (BTC) and small basket cell (SBC), across all layers and targeting all excitatory and inhibitory m-types (see Methods). Moreover, rewiring was restricted to the most central part of the model (Figure 2A, red hexagon); all other connectivity remained unchanged.

Overall, the number of synapses remained relatively constant (∼25 % increase; Table 1), but they were spread over a much larger number of connections (over 700 % increase). Consequently, individual inhibitory connections were formed by a much lower number of synapses per connection (Figure 2B). Note that the low number of synapses per connection is indeed a feature of the electron-microscopic mouse dataset used in (Schneider-Mizell et al., 2023), while the baseline connectome matches the higher mean number of synapses per connection from paired light-microscopic reconstructions in rat (Reimann et al., 2015). We confirmed that synapses from BTC and SBC neurons were mainly targeting inhibitory (15-fold increase) instead of excitatory (two-fold decrease) neurons (Figure 2B, C; Figure S1). This is further validated by looking at the distribution of BTC and SBC synapses on the dendritic morphologies of two exemplary inhibitory and excitatory neurons before and after rewiring (Figure 2D; extensive list of examples in Figure S3 and S4 respectively).

We ran simulation experiments with both the original and the rewired connectome where we activated the rewired BTC and SBC interneurons by means of a constant 1 s current injection (Figure 2E and Methods). During spontaneous activity before applying the current injection, we observed a slight increase

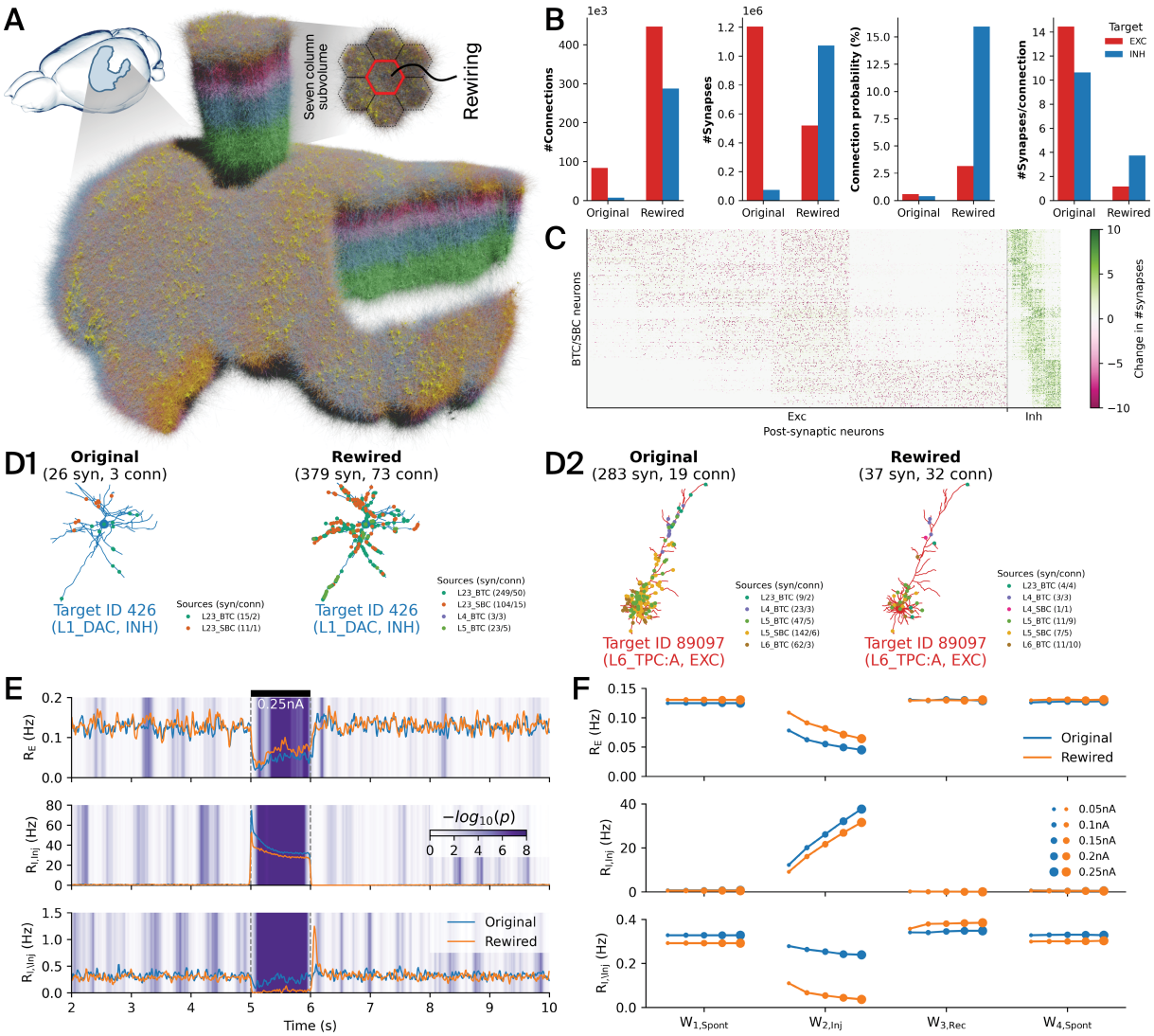

Figure 2: Rewiring of VIP+ interneuron connectivity in a detailed model of rat somatosensory cortex. A: Large-scale, anatomically (Reimann et al., 2022) and physiologically (Isbister et al., 2023) detailed model of the rat non-barrel somatosensory cortex. The central cortical column (red hexagon) of the seven column subvolume was subject to rewiring. **B:** Total number of connections, synapses, mean connection probability and mean number of synapses per connections in the original vs. the rewired connectome between VIP+ (BTC and SBC types) sources and excitatory (red) and inhibitory (blue) targets. **C:** Change in numbers of synapse per connection after rewiring, showing all pairs of 530 source (BTC and SBC types; y-axis) and 30,190 target neurons (26,787 excitatory, 3,403 inhibitory; x axis). Only 10 % of the actual density is plotted. The color scale spans the $\pm 90^{\text{th}}$ percentile of values. **D:** Synapses from different BTC/SBC source types (as indicated by the legend) targeting an exemplary inhibitory (D1) and excitatory (D2) neuron in the original vs. the rewired connectome. Small numbers denote numbers of synapses and connections respectively. More examples in Figure S3 and S4. **E:** Instantaneous firing rates during a current injection experiment (bin size 10 ms; smoothed with Gaussian kernel with SD 1.0) of excitatory (R_E) and inhibitory (injected: $R_{I,Inj}$, non-injected $R_{I,\backslash Inj}$) populations. A constant 0.25 nA current was injected into BTC/SBC types from time point 5 s to 6 s. Background shading denotes significant differences between original and rewired activity, computed as the negative decimal logarithm of the p-value of a Wilcoxon rank-sum test applied on a 200 ms sliding window. **F:** Average firing rates of excitatory and inhibitory populations for all injection currents (0.05 to 0.25 nA) and time windows (see Methods).

Connectome	#Conn	Diff	%	#Syn	Diff	%
Baseline	90,200	—	—	1,276,686	—	—
Rewired	733,742	643,542	713.5	1,593,595	316,909	24.8

Table 1: Connection and synapse counts in rewired interneuron connectivity. The table contains the overall numbers of connections (#Conn) and synapses (#Syn) of the connectome with rewired interneuron connectivity with the respective differences to the original (baseline) connectome, in absolute terms (Diff) and percentages (%). Only the connectivity in the central cortical column originating from BTC/SBC source types which was subject to rewiring is considered here.

in excitatory and slight decrease in inhibitory firing rates in the rewired connectome. During BTC and SBC activation, the firing rates of these types initially increased strongly, followed by a decay to a lower level that was still elevated compared to baseline conditions. The other inhibitory types (that were not injected with current) were largely inhibited by the newly created class of inhibitory targeting interneurons which was not the case in the original connectome. During this period, excitatory neuron firing rates decreased for both connectomes, but more strongly for the original connectome. This means that BTC/SBC interneurons still provided stronger inhibition than disinhibition to excitatory neurons. After the current injection ended, a brief rebound peak in the firing rates of the other inhibitory types was observed for the rewired connectome. The same result holds over a range of injected currents from 0.05 nA to 0.25 nA (Figure 2F).

2.3 Simplified connectivity of a detailed model of rat somatosensory cortex

In a second experiment, we rewired the connectivity between excitatory neurons in the central column of the network model of the rat somatosensory cortex (Figure 2A; red hexagon). Rewiring was done by first simplifying the connectivity to one of the five following stochastic model descriptions (Gal et al. (2020); Figure 3A and Methods), and then generating fixed instances of them as described in the rewire case before (Figure 1C, green).

1st order: Constant connection probabilities between all pairs of neurons.

2nd order: Distance-dependent connection probability between the pre- and post-synaptic neuron.

3rd order: Bipolar distance-dependent connection probability based on two alternative distance-dependent probability functions for the pre-synaptic neuron being axially (i.e., along cortical depth axis) either above or below the post-synaptic neuron.

4th order: Offset-dependent connection probability based on the axial and radial offsets between the pre- and post-synaptic neuron.

5th order: Position-dependent connection probability based on the (absolute) axial position of the pre-synaptic neuron together with the axial and radial offsets between the pre- and post-synaptic neuron.

Important aspects of connectivity, such as synapse locations on dendrites, pathway-specific physiological parameter distributions, and overall numbers of connections were preserved (see Methods).

We compared the structure of all rewired connectomes with the original (baseline) connectome (Figure 3B-D and Methods). As intended, only connections between excitatory neurons were rewired, by adding and deleting connections. The more extreme a simplification was, the fewer connections remained unchanged (Fig 3B, white). No connections from, to, or between inhibitory neurons were changed in any of the connectomes. The number of synapses forming a connection was drawn from pathway-specific distributions derived from the baseline connectome (see Figure S5). At the same time, while the overall number of connections was largely preserved (max. difference 0.032 %), connection counts in individual pathways could shift. Consequently, the total number of synapses could change quite drastically in the

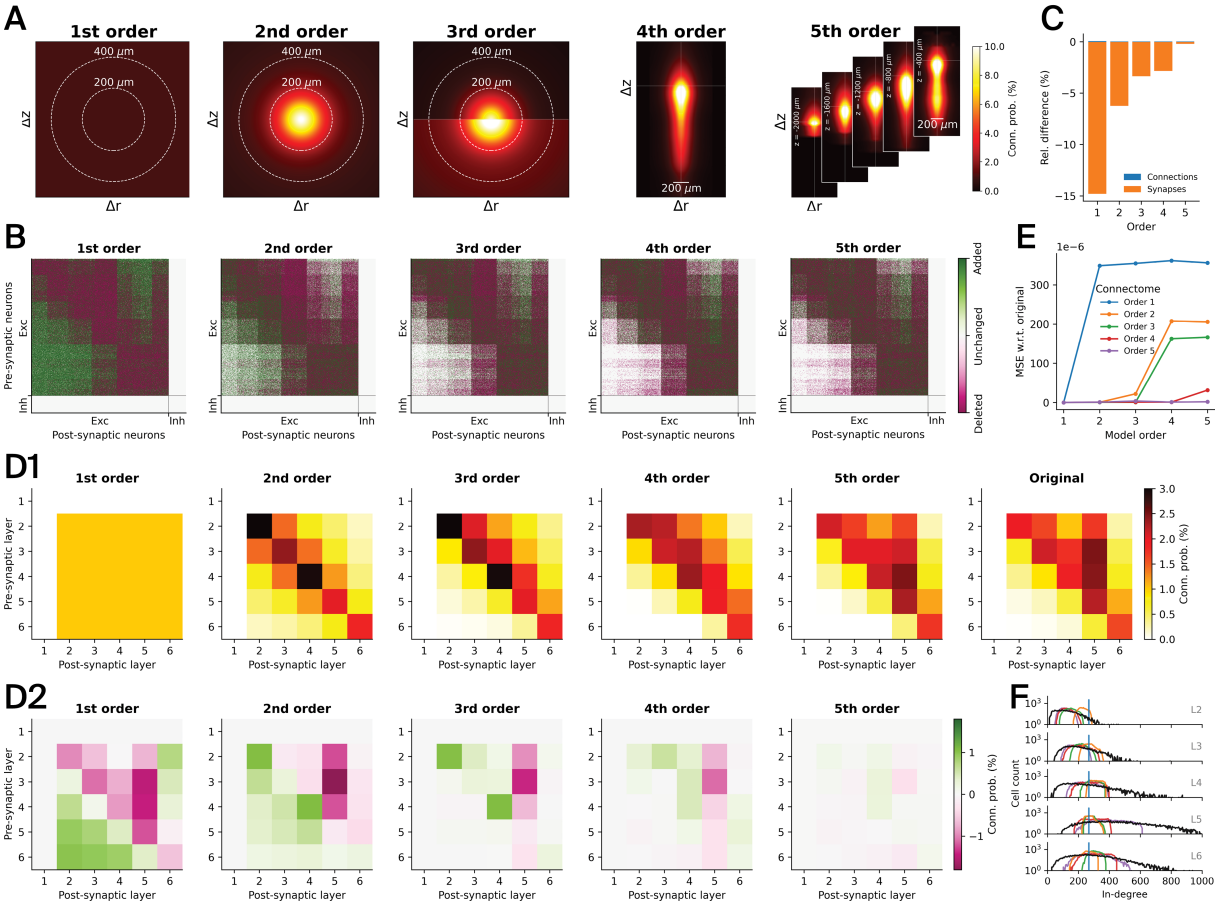

Figure 3: Simplified connectivity of a detailed model of rat somatosensory cortex. A: All connections between excitatory neurons in the central cortical column of the model were rewired based on five simplified stochastic models of connectivity, whose parameters were fitted against the actual connectivity data from the detailed cortical model: 1st order — constant, 2nd order — distance-dependent, 3rd order — bipolar distance-dependent, 4th order — offset-dependent, and 5th order — position-dependent (Δr... radial offset, Δz... axial offset, z... axial position). **B:** Resulting adjacency matrices after rewiring, indicating deleted, added, and unchanged connections relative to the baseline connectome. Neurons are order by cell type (excitatory, inhibitory) and layer. Only 10 % of the actual density is plotted. **C:** Relative differences of the numbers of synapses and connections between excitatory neurons with respect to the baseline connectome (see Table 2 for exact numbers). **D:** Average connection probabilities between excitatory neurons in different layers (D1), and differences to baseline connectome (D2). Note that layer 1 does not contain any excitatory neurons. **E:** Mean squared error of the connection probabilities obtained from the given stochastic models (x-axis) with parameters fitted against the simplified connectomes (as indicated by the legend) vs. fitted against the baseline connectome. **F:** In-degree distributions of rewired connectomes by layer. Same colors as in E, with black representing the original connectome.

simpler connectomes (Figure 3C and Table 2): -15 % difference for the 1st order connectome, decreasing to -0.2 % difference for the 5th order connectome. Importantly, preserving the average number of synapses per connection of individual pathways rather than overall synapse count was necessary to preserve biologically parameterized amplitudes of post-synaptic potentials (PSPs). We further validated the simplifications on the level of layer-wise connection probabilities (Figure 3D). In the 1st order connectome, we indeed found a uniform connection probability distribution between all layers which was very different from the baseline. The 2nd order connectome already captured some of the structure present in the baseline; however, the layer-wise connectivity was completely symmetric and lacked any anisotropy. In the 3rd order and higher connectomes, more and more of the underlying structure was captured, and the difference in connection probability in the 5th order connectome was relatively small. This indicates that simplifying connectivity by taking positions and offsets into account closely approximates, on average, the underlying connectivity structure, in line with Gal et al. (2020). Also, we tested to what degree the simplifications of a given stochastic model order destroyed the structure that is captured by all other orders (Figure 3E and Methods). As expected, we found that a model of a given order fully comprised all models of lower order, i.e., an nth order model is indistinguishable from the original connectome at all orders below or equal n. Conversely, at higher orders, large errors were visible. We further found that at the level of incoming connectivity the structural diversity is lessened in the simplified connectomes (Figure 3F). Finally, we also validated the mean numbers of synapses per connection and important physiological synapse parameters per pathway, i.e., for each pair of pre- and post-synaptic m-types (Figure S5 and Methods). We found that the means and SDs of the pathway-specific parameter distributions from the baseline connectome were largely preserved in the rewired connectomes. Only for axonal delays, we found deviations in some of the pathways. This can be explained by the fact that unlike the other synaptic parameters, axonal delays were not modeled in a pathway-specific way taking morphological differences into account, but by fitting the overall statistics of distance-dependent axonal delays (see Methods).

Connectome	**#Conn**	**Diff**	**%**	**#Syn**	**Diff**	**%**
Baseline	7,203,362	—	—	31,212,240	—	—
1st order	7,205,703	2,341	0.032	26,603,307	-4,608,933	-14.766
2nd order	7,203,361	-1	-0.000	29,272,858	-1,939,382	-6.214
3rd order	7,203,364	2	0.000	30,173,482	-1,038,758	-3.328
4th order	7,203,362	0	0	30,332,403	-879,837	-2.819
5th order	7,203,362	0	0	31,153,461	-58,779	-0.188

Table 2: Connection and synapse counts in rewired simplified connectomes. The table summarizes the numbers of connections (#Conn) and synapses (#Syn) of all simplified connectomes together with the respective differences to the baseline connectome, in absolute terms (Diff) and percentages (%). Only the connectivity between excitatory neurons in the central cortical column of the network model which was subject to rewiring is considered here.

We then ran a series of network simulations in order to quantify functional changes in the simplified connectomes. Specifically, we re-calibrated the five rewired circuits to exhibit *in vivo* like spontaneous activity using the calibration procedure described in Isbister et al. (2023), in order to dissociate the effect of the redistribution of pathway strengths from the effect of higher-order structure. Briefly, the calibration iteratively parameterizes layer-specific conductance injections into the neurons until their firing rates match expected values. These injections represent the extrinsic inputs from regions that are not part of the network model. The expected values were a constant fraction of the firing rates observed in *in vivo* recordings. This fraction, P_{FR}, was set to values below or equal to 1.0 to compensate for the presence of silent and hence "invisible" neurons (Olshausen and Field, 2006; Wohrer et al., 2013; Buzsáki and Mizuseki, 2014). We use the notation LkE and LkI to denote excitatory (E) and inhibitory (I) populations in layer k respectively. Without re-calibration, the spontaneous activity for some simplified connectomes was in a synchronous activity state characterized by bursts of activity throughout all layers except layer 1 (at $P_{FR} = 0.8$; Figure 4A, left). This indicates that inhibition was no longer able to keep recurrent excitation under control. After several calibration iterations, the spontaneous activity was in an asynchronous activity state closely matching the expected firing rates, e.g., Figure 4A (right), for

most values of P_{FR} (Figure 4B). For most simplified connectomes, only three iterations were required (Figure 4C) whereas the 2nd order connectome (distance-dependent connectivity) had remaining errors even after five iterations (Figure 4B, red arrow; Figure S6; Figure S7). This highlights the importance of biologically realistic connectomes in order to obtain *in vivo*-like activity.

As the layer-specific injections represent extrinsic inputs, their changes during re-calibration indicate to what degree a population is driven by intrinsic versus extrinsic populations (Figure 4D). We found that L4E neurons required less conductance injection compared to the baseline connectome, which implies that their activity was shifted towards a more internally driven spontaneous activity regime. This effect was most pronounced in the 2nd order connectome and got successively weaker for more complex models. Moreover, we observed an inverse shift for L6I neurons towards a more externally driven regime, despite the fact that afferent connections to I populations had not been rewired. This effect was most pronounced in the 1st order connectome. For L6E neurons, we observed diverse effects, being more externally driven in 1st and 2nd order, but more internally driven in 3rd to 5th order connectomes. This indicates the importance of the anisotropy of connectivity introduced in the 3rd order connectome. As indicated before (Figure 3F), the incoming connectivity was re-distributed, with some layers receiving more and some layers receiving less synaptic input in the simplified connectomes. As expected, we found that the additional amount of conductance required in all E populations was highly correlated with the change in afferent synapse count (Figure 4E; Pearson correlation coefficient $r = 0.87$; $p < 10^{-6}$ base on two-sided Wald test with t-distribution). This confirms our previous interpretation that the strength of conductance injection is an indication of how much a population is externally versus internally driven. Even though firing rates matched baseline after re-calibration, the correlations of spiking activity were quite different in the simplified connectomes (Figure S8). While correlations in the baseline connectome steadily increased from superficial to deeper layers, the pattern was less clear cut in the simplified connectomes. In the least simplified 5th order connectome correlations were most similar to baseline, but generally lower, especially in layer 6. This change is the result of reduced higher-order structure after the impact of a redistribution of pathway strengths has been controlled for.

3 Discussion

We present in this work a connectome manipulation framework that allows causal structure–function relationships to be studied in a systematic and reproducible way. We have demonstrated its utility in two exemplary applications using a detailed network model of the rat somatosensory cortex (Reimann et al., 2022; Isbister et al., 2023). In one experiment, we increased the inhibitory targeting specificity of VIP+ interneurons based on trends found in mouse EM data. In this case, we employed the framework to transplant specific connectivity provided by an adjacency matrix together with the numbers of synapses per connection in a deterministic way. In another experiment, we decreased the biological realism of the network model and studied the effect of such manipulation by rewiring the connectome based on simplified stochastic connectivity rules. In this case, we utilized the framework to rewire the entire connectivity between excitatory neurons in a stochastic way. Both of these experiment could supposedly have been conducted in an ad-hoc way without use of such framework. However, this would have required us to solve several challenges our framework inherently takes care of. Specifically, it allowed us to fit five simplified stochastic models of connectivity against the baseline connectome (Figure 3A). It enabled us to evaluate them during rewiring in order to generate connectome instances on the connection level (Figure 1D1). It ensured that the synaptic physiology of individual pathways was preserved, by fitting property distributions to the baseline connectome and evaluating them while turning the connectome into a description at the synapse level (Figure 1D2). It preserved the patterns of innervation of dendritic compartments by reusing existing synapse locations for the new connections (Table 3). While for stochastic models of connectivity the exact number of connections is in general not predetermined, the framework provided means to match the numbers to the baseline connectome (Table 2). It provided validations of the rewired connectomes on the connection and synapse level by structural comparison with the baseline (Figure 2B, C; Figure 3B-D; Figure S1, Figure S2, Figure S5). It offered high performance due to its parallel architecture (Figure 5), allowing us to rapidly rewire ∼7M connections and ∼30M synapses in less than 10 minutes on five computation nodes. Finally, it generated new SONATA circuits that could be readily used in simulation experiments (Figure 2E, F; Figure 4). Altogether, this shows that having

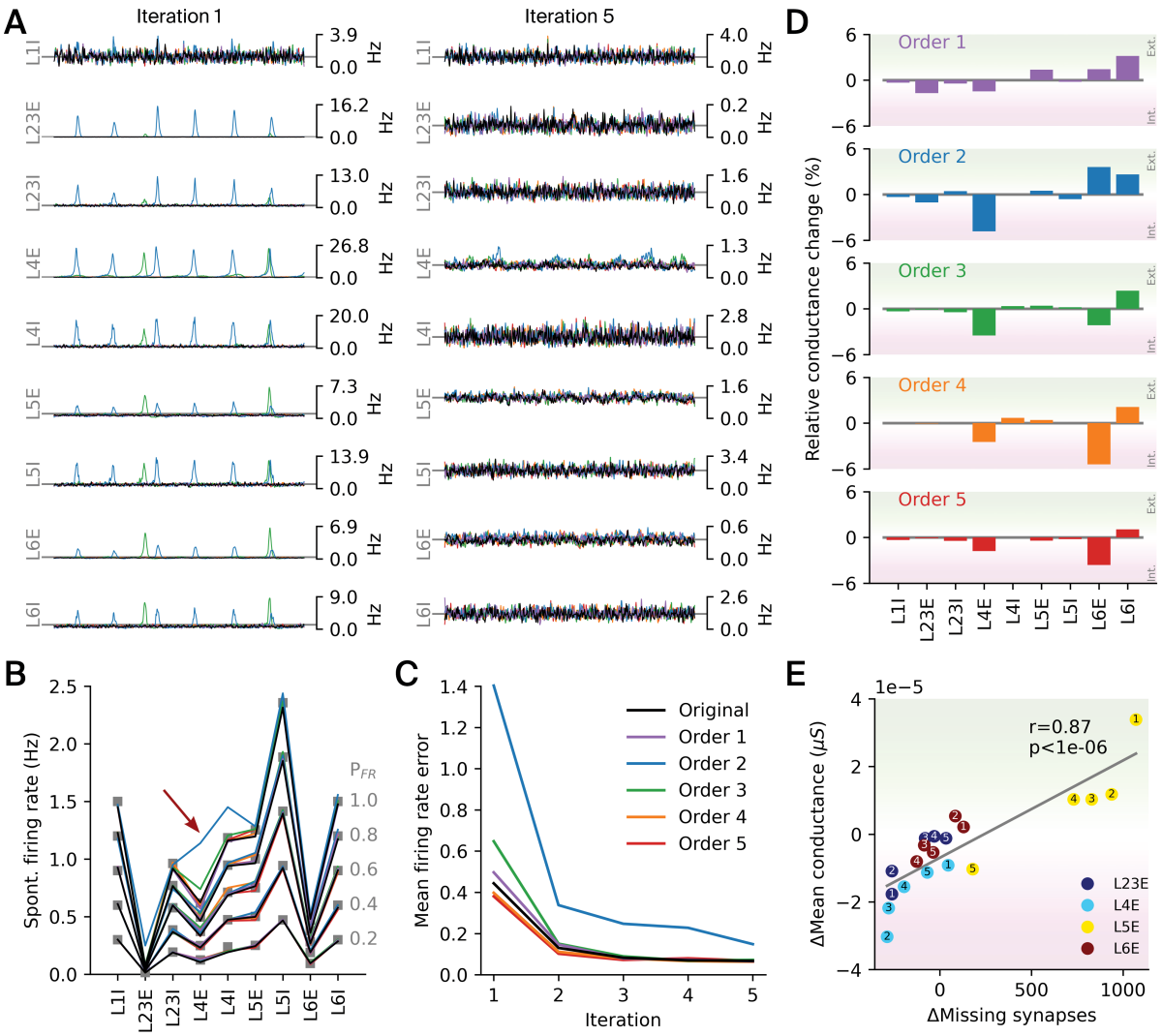

Figure 4: Functional implications of simplified connectivity. **A:** Instantaneous firing rates of 5 s spontaneous activity of E and I populations across layers for each of the rewired connectomes before and after re-calibration, shown for an *in vivo* fraction $P_{FR} = 0.8$ (cf. Figure S7 with $P_{FR} = 1.0$). **B:** Spontaneous firing rates after re-calibration (five iterations) closely matching *in vivo* references (grey squares) for most values of P_{FR} (see Methods), except for 2nd order connectome at $P_{FR} = 1.0$ (red arrow). **C:** Mean firing rate error, computed as the Euclidean distance to the reference rates and averaged over all P_{FR} values, for five calibration iterations. **D:** Changes in mean conductance injection relative to baseline after re-calibration, indicating neuron populations shifted towards a more externally (positive change) or internally (negative change) driven regime through rewiring. **E:** Mean conductance injection versus number of missing synapses (both relative to baseline) for E populations across layers, as indicated by the legend. Small numbers denote the orders (1 to 5) of the rewired connectomes.

such a framework as a reference tool greatly helps to standardize and reproduce *in silico* manipulation experiments.

In our two exemplary experiments the framework allowed us to make several predictions. We predict that during spontaneous activity, activation of inhibitory targeting VIP+ interneurons reduces the firing rate of excitatory populations even though only 33 % of the synapses are targeting excitatory neurons (Figure 2). That is, although this group mostly shuts down firing of other interneurons the effect of this disinhibition is still weaker than its direct inhibition. Additionally, we learned that the removal of higher-order structure of connectivity when only the distance-dependent trends remained led to a transitions from an asynchronous to a synchronous state which had to be compensated for by reducing the amount of excitation from extrinsic sources (Figure 4). This was especially evident in layer 4, indicating that the role of layer 4 as input layer (Douglas and Martin, 2004; Miller, 2016) may be tied to the higher-order structure of recurrent connectivity. Structurally, the reduction or removal of higher-order structure led to a redistribution of strengths of layer-specific pathways. This is expected, as neuronal networks with more complex structure are associated with long-tailed degree distributions, while simplified networks have more homogeneous distributions. But it highlights the difficulty of dissociating first-order statistics, such as pathway strengths, from higher-order statistics in morphologically detailed models. Additionally, we found that in less realistic connectomes it may be harder to achieve a biologically realistic state (Figure S7), indicating the importance of properly modeling the higher-order connectivity structure. Finally, differences in the complexity of higher-order structure led to differences in the laminar patterns of spiking correlations.

Even though our two experiments were just brief outlines of full-scale experiments, we already gained interesting insights. In the future, such experiments could be done more extensively by direct use of EM reconstructions (MICrONS Consortium et al., 2021; Winding et al., 2023). For this to work, one would first need to create a SONATA version of an EM dataset, which involves modeling both the set of neurons and the connectivity between them. Then, one can study it by creating simplified but otherwise equivalent versions of the connectome as we have done before. This would provide insights into the importance of higher-order structure in a biologically measured and not merely predicted connectome. Also, the importance of individual pathways can in principle be studied by simply removing them, but such a severe manipulation may not be too insightful. A more fine-grained control for degrading them is provided by use of our manipulation framework. Recently, the higher-order network structure has been causally linked to important functional properties, such as reliability, efficiency, and population coupling (as defined by Okun et al. (2015)), using fine-grained and targeted manipulations implemented in our connectome manipulation framework (Egas Santander et al., 2024). We believe that the full potential of EM reconstructions can only be developed by following such ideas and going from purely correlational metrics to actual causation. The need for investigating causal interaction in order to understand brain function has been pointed out before (Reid et al., 2019).

As our framework is inherently equipped to interpret connectivity at different levels of abstraction (Figure 1C), it can be used to bridge the scales between more and less detailed network models. For example, it could be employed to wire a detailed network model according to architectures used in machine learning in order to study under what conditions they would lead to biologically realistic results, thereby tightening the loop between neuroscience and artificial intelligence (Aru et al., 2023; Gershman, 2024; Gopinath, 2023; Surianarayanan et al., 2023; Verzelli et al., 2024). Also, by use of the open SONATA format (Dai et al., 2020), our framework supports not only biologically detailed but also point neuron network models, allowing for example the transformation of a detailed into a point neuron network model. Rössert et al. (2017) describes the adjustments to synaptic parameters required for such an endeavor that could be easily implemented as part of the operations described in Figure 1D2 in the context of the transplant functionality (Figure 1C, blue). Together with the recent publications of openly available large-scale models (Billeh et al., 2020; Dura-Bernal et al., 2023; Isbister et al., 2023), this makes our work also relevant for the point neuron model community.

Taken together, our framework serves as a flexible starting point for manipulating connectomes in a systematic and reproducible way, which can be easily extended and adapted to individual use cases. New code modules can be simply integrated into the existing framework, such as new types of stochastic

models, tools for fitting them against existing data, new manipulation operations, more specific synapse placement rules, and additional structural validations. Together with the ability to actually simulate such manipulated connectomes, this represents a powerful tool for fully understanding the role of connectivity in shaping network function.

4 Methods

4.1 Details of the connectome manipulation framework

4.1.1 Stochastic models and model building

A *model* in the context of this framework is a simplified (stochastic) representation of certain aspects of connectivity required by some of the manipulation functions. Models can be stored as JSON (JavaScript Object Notation) files and are thus human-readable and editable; some models containing large amounts of data have them stored in an additional HDF5 (Hierarchical Data Format) file for easy machine processing. The process of *model building* refers to fitting models against existing connectomes and producing a model file. Some generic model formats and extensions to existing models also exist, allowing for a more fine-grained control. More details about the available models and tools for model fitting can be found in Table S1 and S2 respectively.

4.1.2 Algorithmic steps for generating new connection

For generating new synaptic connectivity, two algorithmic steps are involved. In the first step, connectivity assignment (Figure 1D1), the incoming connectivity from pre-synaptic neurons to each post-synaptic neuron is assigned. This is done by determining the connection probability p_i of all potential pre-synaptic source neurons N_i to be connected with a post-synaptic target neuron N_{Tgt}. The connection probabilities are obtained from a deterministic or stochastic connection probability model (Table S1). Then, new source neurons are randomly sampled from all N_i according to p_i, optionally preserving the in-degree (Table 3). If newly drawn connections already existed in the original connectome, they can optionally be kept unchanged; otherwise, they will be replaced by reusing other existing or generating new connections (Table 3). In case the provided connection probability model returns only probability values zero and one, the resulting connectivity assignment will be deterministic.

In the second step, synapse assignment and parameterization (Figure 1D2), synapses on the post-synaptic dendrite are placed, parameterized, and (randomly) assigned to form all incoming connections as determined in the first step, unless existing connections are kept or reused (Table 3). New synapses can be placed by either duplicating existing synapse positions, randomly generating new positions on the dendrite, or loading positions externally. The number of synapses to assign to a new connection, as well as their physiological parameter values, can be sampled from existing connections or are given by stochstic models defining their (pathway-specific) parameter distributions (Table S1). Optionally, the number of synapses per connection can also be provided deterministically through a synaptome matrix.

4.1.3 Operation principle of connectome manipulations

Connectome manipulations follow a block-based operation principle and can be run either serially or in parallel on multiple computing cores or nodes (Figure 5). This works by splitting the input SONATA connectome post-synaptically into disjoint edges tables each of which contains all synapses targeting a block of post-synaptic neurons, the size of which can be configured. Each edges table is then manipulated independently, and the output is written to a separate .parquet file (i.e., in Apache Parquet format). After all manipulations are completed, the individual .parquet files can be kept for further processing, and/or can be merged to a single output SONATA connectome using the `parquet-converters` utilities (external dependency). Note that the same operation principle also applies for wiring an empty connectome from scratch, in which case all input edges tables will be initialized empty and each manipulation will add new edges to them.

All manipulation operations are implemented in separate Python code modules, which makes it easy to add new functionality to the framework. A variety of connectome manipulation operations already exists for different use cases, as summarized in Table S3.

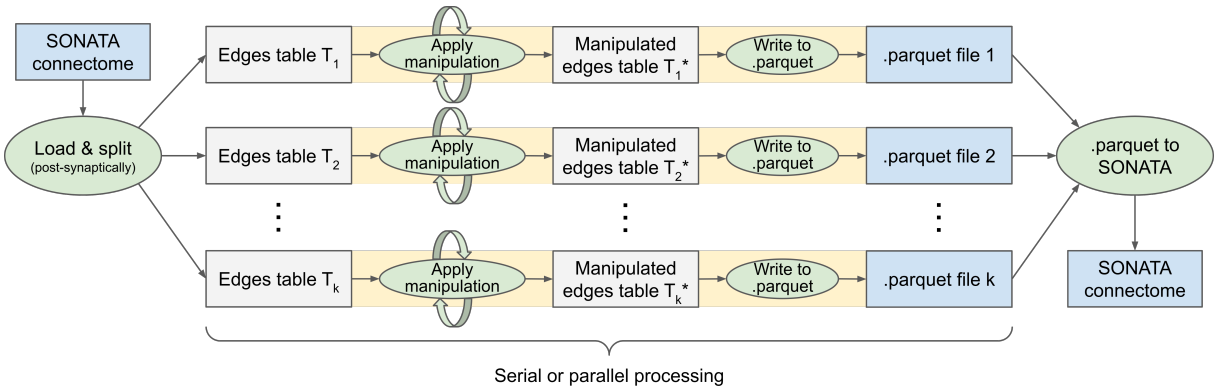

Figure 5: Operation principle of connectome manipulations. Internally, a connectome manipulation works by splitting the input SONATA connectome by post-synaptic neurons into disjoint edges tables. Each table is manipulated independently, and the output is written to a separate .parquet file. This enables operations to be run in series or in parallel. After all manipulations are completed, the individual .parquet files are merged to a single output SONATA connectome using `parquet-converters` (external dependency).

4.1.4 Matching the total number of connections in rewired connectomes

When fitting stochastic connection probability models (as in Table S1) against existing connectomes and using them for rewiring, the resulting number of connections in the rewired connectome will in general not exactly match the number of connections in the original connectome for three reasons. First, stochastic connection probability models are in general simplified descriptions of connectivity which may not capture the underlying shape of the actual connection probabilities exactly (e.g., approximating distance-dependent connectivity by an exponential function). Second, stochastic connectivity models are not evaluated globally but locally for independently drawing incoming connections for each post-synaptic neuron (see first algorithmic step in Figure 1D1), allowing for efficient parallel processing (see Figure 5). Third, because of the stochastic nature of such probability models, the exact number of resulting connections varies in different random instances.

For applications where matched numbers of connections are desirable, our framework provides functionality for matching the overall number of connections as close as possible to the original number, based on the following assumptions. Rewiring is done by computing connection probabilities p_{ij} for all pre-synaptic neurons i to be connected to a given post-synaptic neuron j based on a stochastic model of connectivity. A specific instance (realization) randomly drawn from p_{ij} for each post-synaptic neuron j has $N_{in,j}$ incoming connections. Thus, the expected number of incoming connections of a post-synaptic neuron j on average can be computed as

$$\overline{N}_{in,j} = \text{ROUND}\left(\sum_i p_{ij}\right) \tag{1}$$

For adjusting the resulting number of connections, we implemented a global probability scaling factor p_{scale} into the rewiring operation which scales the connection probability p'_{ij} given by a stochastic connectivity model, i.e., $p_{ij} = p'_{ij} \cdot p_{scale}$ (by default, $p_{scale} = 1.0$). Using such global factor has the advantages that it neither changes the overall shape of the probability function nor introduces any dependencies between parallel processes or biases depending on the number of data splits.

In order to rapidly predict the total number of connections on average (i.e., independent of the random seed) when using a given connection probability model (incl. scaling factor p_{scale}), we also implemented an "estimation run" option as part of the rewiring operation (see Table S4). This options allows a rewiring operation to be executed with early stopping, without generating an actual connectome or output file, but writing the average number of incoming connections $\overline{N}_{in,j}$ according to Eq. 1 into a data log file. Using these values from the data log, the total number of connections can be computed as sum over all

post-synaptic neurons j, i.e.,

$$\overline{N} = \sum_j \overline{N}_{in,j} \tag{2}$$

Based on this number, a scaling factor p_{scale} can be computed by

$$p_{scale} = N_{orig}/\overline{N} \tag{3}$$

where N_{orig} is the total number of connections in the original connectome.

While the above functionality allows the adjustment of the resulting numbers on average, a single random instance may still deviate. Therefore, we implemented yet another option for optimizing the drawn number of connections in a single instance. Specifically, the number of incoming connections for each post-neuron will be optimized to match its average number of connections. This is done by repeating the random generation up to 1,000 times and keeping the instance with the number of connections exactly or as close as possible matched to the average.

So, for closely matching the overall number of connections, we propose the following semi-automatic two-step procedure:

Step 1: Matching the mean

- Run estimation runs iteratively, until the total predicted number of connections $\overline{N}$ in the rewired connectome (Eq. 2) exactly matches the total number of connections N_{orig} in original connectome. A few iterations are usually required since a non-linear rounding operation to integer numbers of connections is involved (see Eq. 1).
- After each iteration, compute a new scaling factor $p_{scale,new}$ using Eq. 3.
- Update $p_{scale} = p_{scale} \cdot p_{scale,new}$ based on this new estimate, to be used in the next iteration.
- Convergence is reached if $p_{scale,new} = 1.0$, i.e., the expected number of connections exactly matches the number in the original connectome. However, convergence is not guaranteed and values may oscillate around the theoretical optimum $\overline{N}_{opt} = \sum_j \sum_i p_{ij}$ (i.e., without rounding). In such case, the value closest to the optimum should be used.

Step 2: Matching an instance to the mean

- Use converged (or closest) value p_{scale} from step 1.
- Enable option to optimize numbers of connections in a single instance.
- Run actual rewiring operation for a specific random seed (i.e., no estimation run).

Even though this procedure usually finds a close solution after only a few iterations, it is not guaranteed to converge to the exact number under all circumstances. This is mainly due to the discrete nature of the local rewiring where integer numbers of incoming connections are independently drawn for each post-synaptic neuron, and the parallel processing architecture which limits the exchange of information, such as the actually drawn numbers of connections, among independently processed data splits.

4.1.5 Preserving aspects of connectivity in rewired connectomes

Certain aspects of connectivity can be preserved when rewiring an existing connectome (i.e., by use of `conn_rewiring`, see Table S3), such as entire connections, synapses per connection, physiological parameters of synapses, synapse positions on the dendrite, and in-degrees, as summarized in Table 3. Since not all rewiring options are independent of each other, the table also indicates which combinations of options are actually meaningful in order to preserve certain properties. A detailed description of all rewiring options can be found in Table S4. An important special case of rewiring which is also supported is to wire an empty connectome from scratch, which only involves creating new connections. Hence, none of the rewiring options involving existing synapses are applicable in this case.

Table 3: Preserved aspects of connectivity in rewired connectomes. The table shows different groups of options (Table S4) regarding existing connections (blue shaded), synapse physiology (green shaded), synapse positions (red shaded), and in-degree (yellow shaded) that can be used for rewiring (i.e., by use of `conn_rewiring`) and how they affect certain aspects of connectivity in the rewired connectome. The table entries indicate whether or not a given property is preserved for a post-synaptic neuron (Y: yes, N: no), or if it is independent of a certain option (n/a: not applicable).

	Preserved properties				
Option	Conn. source	Synapse per conn.	Synapse physiol.	Synapse position	In-degree
Connections					
`keep_conns`*	Y	Y	Y	Y	n/a
`reuse_conns`*	N	Y	Y	Y	n/a
Otherwise	N	n/a	n/a	n/a	n/a
Synapse physiology					
`sample`	n/a	N	N	n/a	n/a
`randomize`	n/a	N	N	n/a	n/a
Synapse positions					
`reuse`	n/a	n/a	n/a	Y	n/a
`random`	n/a	n/a	n/a	N	n/a
`external`	n/a	n/a	n/a	N	n/a
In-degree					
`keep_indegree`	n/a	n/a	n/a	n/a	Y
Otherwise	n/a	n/a	n/a	n/a	N

*If both are selected, `keep_conns` will override `reuse_conns`.

4.1.6 Structural comparison

The connectome manipulation framework includes additional tools for *structural comparison* which allow a user to compare connectivity-related properties of two connectomes. Specifically, they allow the comparison of the baseline with a manipulated connectome with regard to differences in their connectivity structure and their distributions of synaptic properties, together with visualizations on the single-neuron level or grouped by populations of neurons. More details about the available tools for comparing connectomes can be found in Table S5.

4.2 Detailed network model of the rat somatosensory cortex

We employed connectome manipulations in a detailed anatomical (Reimann et al., 2022) and physiological (Isbister et al., 2023) network model of the non-barrel rat somatosensory cortex (Figure 2A). While the full model contains over 4M neurons, we utilized the openly available 1.5 mm diameter subvolume composed of seven hexagonal columns, which has been released under DOI 10.5281/zenodo.8026353. This data-driven subvolume model consists of over 210k biophysically detailed neurons belonging to 60 different morphological types (m-types) and 208 different morpho-electrical types, which are connected by over 400M synapses with probabilistic transmitter release and 5 distinct forms of short-term dynamics of depression and facilitation. The connectome is based on axo-dendritic appositions, which has been demonstrated to reproduce non-random trends identified in biological networks (Gal et al., 2017). As depicted in Figure 2A, we applied connectome manipulations only to connections between neurons within the central cortical column of the seven column subvolume, in order to avoid potential edge effects. For all manipulations with geometry-dependent connection probabilities (i.e., 2nd to 5th order), we used the flat coordinate system mapping released in Bolaños-Puchet et al. (2024) under 10.5281/zenodo.10686776, with the x/y-axes parallel to the cortical layers, and the z-axis along the cortical depth.

For simulating the model, we used the CoreNEURON simulator (Kumbhar et al., 2019) together with the openly-available Neurodamus simulator control application (see Software and data availability). As

for manipulations, we only simulated neurons within the central cortical column of the seven column subvolume, using connections from the baseline or one of the manipulated connectomes between them. The neurons themselves were not subject to any manipulations and remained identical in all simulations.

In order to obtain *in vivo*-like spontaneous activity during simulations, we compensated for missing excitatory inputs that were external to the network model, as described in Isbister et al. (2023). Input compensation was given by statistically independent, population-specific somatic conductance injections from Ornstein-Uhlenbeck (OU) processes that would mimic aggregated random background synaptic inputs. The compensation mechanism was based on three meta-parameters: the extracellular calcium concentration Ca, the fixed ratio between standard deviation and mean of the underlying OU processes R_{OU}, and a constant fraction P_{FR} of the population-specific *in vivo* reference values (Reyes-Puerta et al., 2015; De Kock et al., 2007), taking into account that extracellularly recorded firing rates are known to be overestimated to an unknown degree (Olshausen and Field, 2006; Wohrer et al., 2013; Buzsáki and Mizuseki, 2014). Unless noted otherwise, we used the calibration of OU-parameters from Isbister et al. (2023) on the full seven column subvolume for Ca of 1.05 mM, a ratio R_{OU} of 0.4, and an *in vivo* proportion P_{FR} of 0.3.

4.3 Details of rewired VIP+ interneuron connectivity

4.3.1 Interneuron rewiring

The baseline connectome of the network model of the rat somatosensory cortex (Reimann et al., 2022; Isbister et al., 2023) is based on the detection of axo-dendritic appositions as potential synapses, followed by a target-unspecific pruning step that prefers multi-synaptic connections (Reimann et al., 2015). In order to introduce a preference for VIP+ interneurons to target other inhibitory neurons, we pruned the original (unpruned) set of appositions originating from VIP+ neurons (i.e., bitufted and small basket cells; m-types BTC and SBC) in layers 2/3, 4, 5, and 6 such that 96.5 % of their potential synapses on non-inhibitory target neurons were removed (Reimann et al., 2022), thereby reproducing inhibitory targeting trends found in MICrONS data (Schneider-Mizell et al., 2023). Based on this pruning rule, we extracted new adjacency and synaptome (i.e., numbers of synapses per connection) matrices for the connectivity within the central cortical column, as well as the exact synapse positions on the dendrite, which were stored in generic model formats of types `LookupTableModel` (adjacency, synaptome) and `PropsTableModel` (positions) respectively (see Table S1).

In addition, another two model descriptions were required to run rewiring: a stochastic model for realizing new connections by forming and parameterizing synapses, and a stochastic model for assigning their axonal delays. For realizing connections, a connection properties model of type `ConnPropsModel` (see Table S1) was fitted against the central column of the baseline connectome using the `conn_props` code module (see Table S2), by extracting pathway-specific parameter distributions, i.e., for all 8×60 pairs of pre- and post-synaptic m-types that were subject of rewiring, of the distribution types as summarized in Table 4A. The numbers of synapses per connections were not drawn from that model, but provided deterministically through the synaptome. All remaining properties not listed in this table were not relevant for simulations and were set to zero. Pathways with less than 10 connections were treated as missing values and therefore gradually interpolated from similar pathways (as detailed in Table S2). For pathways with more than 10k connections, a random subset of 10k connections was used for distribution fitting.

Table 4: Connection property distributions. Pathway-specific property distribution types that were fitted against the baseline connectome and used for drawing property values when realizing connections in (A) interneuron rewiring and (B) simplified connectomes.

	Distribution type		
Property	**(A) Interneuron**	**(B) Simplified**	**Description**
`n_syn_per_conn`	—	discrete (integer)	Synapses per connection
`conductance`	gamma	gamma	Peak conductance (nS)
`conductance_scale_factor`	constant	constant	Conductance scaling factor
`decay_time`	truncated normal	truncated normal	Decay time constant (ms)

`depression_time`	gamma	gamma	Time constant for recovery from depression (ms)
`facilitation_time`	gamma	gamma	Time constant for recovery from facilitation (ms)
`n_rrp_vesicles`	discrete (integer)	discrete (integer)	Number of vesicles in readily releasable pool
`syn_type_id`	discrete (integer)	constant (integer)	Synapse type ID
`u_syn`	truncated normal	truncated normal	Utilization of synaptic efficacy
`u_hill_coefficient`	discrete	constant	Utilization scaling coefficient with calcium concentration

A model for assigning linearly distance-dependent axonal delays of type `LinDelayModel` (see Table S1) was fitted against connections between the relevant pre- and post-synaptic m-types in the central column of the baseline connectome using the `delay` code module (see Table S2). We utilized a distance bin size of 50 μm and did not distinguish between individual pathways.

We then rewired the connectivity originating from BTC and SBC interneurons in the central cortical column using the `conn_rewiring` code module (see Table S3, 3, and S4). We employed the `LookupTableModel` storing the adjacency matrix as deterministic connectivity description, i.e., containing only connection probability values of zeros and ones as determined by the matrix. The numbers of synapses for each individual connections were taken from the `LookupTableModel` storing the synaptome matrix, and the exact synapse positions on the dendrites were externally loaded from the above-mentioned `PropsTableModel` (i.e., `external` option). We did not keep or reuse connections or their physiological parameterization in case they had already existed in the baseline connectome. Instead, pathway-specific physiological parameter values were independently drawn for new connections from the before-fitted connection properties model (i.e., `randomize` option). Likewise, synaptic delays were drawn from the before-fitted axonal delay model depending on the Euclidean distance between the pre-synaptic soma and the synapse position on the post-synaptic dendrite. The rewiring run was launched in parallel on five nodes of a computing cluster using 500 data splits (see Section 4.1.3).

4.3.2 Structural comparison with the baseline connectome

We ran different types of structural comparisons of the baseline and the rewired connectome. First, we compared the synaptome matrices of connections originating from BTC and SBC interneurons in the central cortical column using the `adjacency` code module (see Table S5), results of which are shown in Figure 2B. Second, we extracted the mean connection probabilities and numbers of synapses per connection grouped by m-types using the `connectivity` code module, results of which can be found in Figure S1. Finally, we validated the physiological property distributions grouped by m-types, by extracting their means and standard deviations using the `properties` code module. Important physiological properties that were extracted were `conductance`, `decay_time`, `delay`, `depression_time`, `facilitation_time`, `n_rrp_vesicles`, and `u_syn`, results of which can be seen in Figure S2, together with the numbers of synapses per connection.

4.3.3 Current injection experiment

We ran simulation experiments of the central cortical column of both the baseline and the rewired connectome during which we activated the BTC and SBC interneurons by injecting a constant current. In each simulations, we injected one of the five current strengths 0.05 nA, 0.1 nA, 0.15 nA, 0.2 nA, and 0.25 nA respectively. The total simulation duration of 10 s was divided into four time windows in which the network activity was analyzed afterwards:

$\boldsymbol{W_{1,Spont}}$ ... Spontaneous activity, from $t = 2$ s to 5 s

$\boldsymbol{W_{2,Inj}}$ ... Current injection time window, from $t = 5$ s to 6 s

$\boldsymbol{W_{3,Rec}}$ ... Recovery time window after injection, from $t = 6$ s to 7 s

$\boldsymbol{W_{4,Spont}}$ ... Spontaneous activity, from $t = 7$ s to 10 s

For analyzing the activity, we computed the average firing rates R over three distinct populations of neurons:

$\boldsymbol{R_E}$ ... Excitatory neurons

$\boldsymbol{R_{I,Inj}}$ ... Injected inhibitory neurons (BTC and SBC m-types)

$\boldsymbol{R_{I,\setminus Inj}}$ ... Non-injected inhibitory neurons (i.e., all remaining inhibitory m-types)

Instantaneous population firing rates as shown in Figure 2E were estimated with a bin size of 10 ms and smoothed with a Gaussian kernel with a standard deviation of 1.0. Significant differences between baseline and rewired activity were computed as the negative decimal logarithm of the p-values obtained by a Wilcoxon rank-sum test applied on 200 ms sliding windows of the instantaneous firing rates. Also, average firing rates of all time windows and populations were computed for all current strengths, results of which can be seen in Figure 2F.

4.4 Details of simplified connectomes

4.4.1 Simplified stochastic models of connectivity

We fitted five simplified stochastic connection probability models from 1st to 5th order (Gal et al., 2020) against the connectivity between excitatory neurons in the central cortical column of the baseline connectome using the `conn_prob` code module (see Table S2). The resulting 1st order model was of type `ConnProb1stOrderModel` (with $p_{const} = 0.010$; see Table S1). As all higher-order models depend on geometry, they required correct alignment of the coordinate axis with the cortical layers. We therefore employed a coordinate transformation of the neuron positions to a flat coordinate system, by linearly interpolating the voxel-based flat and depth coordinates of the network model (see Software and data availability) using the `pos_mapping` code module (see Table S2). We scaled the x/y-axis by a factor of $34.0 \cdot 189.0$ from normalized units to μm, and the z-axis by a factor of -1.0, i.e., along the negative cortical depth (in μm). The resulting position mapping extension of type `PosMapModel` (see Table S1) was then applied when fitting connectivity models and using them in manipulations.

For fitting the 2nd and 3rd order models, we used a distance bin size of 50 μm. The resulting connectivity models were of types `ConnProb2ndOrderComplexExpModel` (with $\alpha_p = 0.084$, $\beta_p = 0.000186$, $\gamma = 1.735$, $\alpha_d = 0.017$, $\beta_d = 0.002$) and `ConnProb3rdOrderComplexExpModel` (with $\alpha_{p-} = 0.087$, $\beta_{p-} = 0.000042$, $\gamma_- = 2.0$, $\alpha_{d-} = 0.024$, $\beta_{d-} = 0.001$, $\alpha_{p+} = 0.081$, $\beta_{p+} = 0.001004$, $\gamma_+ = 1.444$, $\alpha_{d+} = 0.013$, $\beta_{d+} = 0.003$) respectively (see Table S1). Unlike the simplified models proposed in Gal et al. (2020), we used reduced, radial symmetric versions of the 4th and 5th order models which had a radial component within in the x/y-plane and an axial component along the z-axis, and position-dependence in the 5th order model only along the z-axis. For constructing these models, we used a radial offset binning from 0 to 450 μm and an axial offset binning from -1550 to 650 μm in steps of 50 μm, and for the 5th order model in addition a position binning from -2400 to 200 μm in steps of 200 μm. The resulting connectivity models were based on bi- and tri-linear interpolation on the regular grids of data bins and were of types `ConnProb4thOrderLinInterpnReducedModel` and `ConnProb5thOrderLinInterpnReducedModel` respectively (see Table S1). We excluded bins with less than 100 data points in order to reduce noise in all model fits.

Again, stochastic models for parameterizing new connections and assigning axonal delays were required. We used the same types of models, a `ConnPropsModel` and a `LinDelayModel`, as for interneuron rewiring (see Section 4.3.1) but which were fitted against the connectivity between 18×18 m-types of excitatory neurons in the central cortical column here. The property distribution types as summarized in Table 4B were used, including a distribution for numbers of synapses per connection. All other model fitting parameters were kept the same as for VIP+ interneuron rewiring.

4.4.2 Rewiring based on simplified stochastic models of connectivity

We rewired the whole connectivity between excitatory neurons within the central cortical column of the baseline connectome using the `conn_rewiring` code module (see Table S3, 3, and S4) based on the stochastic 1st to 5th order connectivity models that had been fitted against the baseline connectome beforehand. For each of the connectivity models, we did this in an iterative way in order to match the total number of connections in the rewired connectomes to the baseline, using the procedure detailed in

Section 4.1.4. Specifically, we ran between three and eight iterations in order to obtain close or exact matches of the overall numbers of connection as summarized in Table 2. Again, each rewiring run was launched in parallel on five nodes of a computing cluster using 500 data splits (see Section 4.1.3).

Importantly, when applying the stochastic connectivity models (i.e., drawing connections in the connectivity assignment step, see Figure 1D1) we employed the same mapping of neuron positions to a flat coordinate system that was used when fitting the connectivity models. When realizing new connections trough individual synapses (i.e., synapse assignment and parameterization step, see Figure 1D2), we reused existing synapse positions on the dendrites (i.e., `reuse` option), but we did not keep or reuse in-degrees, the numbers of synapses per connection or their physiological parameterization. Instead, pathway-specific physiological parameter values were independently drawn from parameter distributions (see Table 4B) given by the connection properties model that had been fitted against the baseline connectome beforehand (i.e., `randomize` option). Likewise, synaptic delays were drawn from the axonal delay model that had been fitted beforehand depending on the Euclidean distance between the new pre-synaptic soma and the newly assigned synapse position on the post-synaptic dendrite.

4.4.3 Structural comparison with the baseline connectome

Again, we ran different types of structural comparisons of the baseline and each of the rewired simplified connectomes. First, we compared the adjacency matrices of the central cortical column using the `adjacency` code module (see Table S5), results of which are shown in Figure 3B. Second, we compared the mean connection probabilities between excitatory neurons within the central column grouped by layers using the `connectivity` code module, results of which can be seen in Figure 3D. Third, we extracted the numbers of synapses per connection grouped by m-types, again using the `connectivity` code module. Finally, we validated the physiological property distributions grouped by m-types, by extracting their means and standard deviations using the `properties` code module. Important physiological properties that were extracted were `conductance`, `decay_time`, `delay`, `depression_time`, `facilitation_time`, `n_rrp_vesicles`, and `u_syn`, results of which can be seen in Figure S5, together with the numbers of synapses per connection.

4.4.4 Model order validation of the simplified connectomes

For validating the model order of the rewired simplified connectomes, we re-fitted each of the stochastic 1st to 5th order models against each of the five simplified connectomes, resulting in 25 model fits whose connection probabilities were given by $p_m^c(x)$ for model order m and simplified connectome order c; x denotes the set of respective input parameters an order-m model depends on, i.e., no input for 1st order, distance variable for 2nd order, etc. For each model order m, we probed the corresponding probability function in steps of $10\,\mu$m of their respective input variables (i.e., distance, offset, position, etc.) and computed the mean-squared error (MSE) with respect to the probabilities $p_m^{BL}(x)$ probed at the same input values of the model fits to the baseline connectome, i.e., the ones that had been used for rewiring in the first place. The MSE for a simplified connectome with order c was computed as

$$MSE_m^c = \frac{1}{N_x} \sum_x \left(p_m^c(x) - p_m^{BL}(x) \right)^2 \tag{4}$$

with N_x being the total number of probed input values of x for a given model order m. Specifically, we probed the probability functions at input values as follows: For $m = 1$, we used $N_x = 1$ since p_1^c is a constant value without any dependencies; for $m = 2$, we used $N_x = 251$ distance values from 0 to $2500\,\mu$m; for $m = 3$, we used $N_x = 501$ (bipolar) distance values from -2500 to $2500\,\mu$m; for $m = 4$, we used $N_x = 51 \cdot 221$ as given by 51 radial offset values from 0 to $500\,\mu$m and 221 axial offset values from -1600 to $600\,\mu$m; for $m = 5$, we used $N_x = 51 \cdot 221 \cdot 201$ as given by the same offsets as before and in addition 201 axial position values from -2000 to $0\,\mu$m. The resulting MSEs for all model orders and connectomes are shown in Figure 3E.

4.4.5 Re-calibration of the simplified circuits

We employed the iterative calibration procedure developed in Isbister et al. (2023) in order to calibrate population-specific OU-parameters (see Section 4.2) in the five circuits with rewired simplified connectomes so that their activity within the central cortical column would exhibit *in vivo*-like spontaneous

activity. Likewise, we also re-calibrated the original circuit with baseline connectome which had an initial calibration on the full seven column subvolume. Specifically, for each of the circuits, we ran five calibration iterations starting from the same initial calibration as the original circuit and using a calcium level Ca of 1.05 mM, a ratio R_{OU} of 0.4, and an *in vivo* proportion P_{FR} from 0.1 to 1.0 in steps of 0.1. After re-calibration, we quantified the mean conductance injections required in the different excitatory and inhibitory populations and layers relative to the original circuit, results of which can be found in Figure 4D.

4.5 Software and data availability

All software and data related to this article are openly available as follows:

Connectome-Manipulator: Python framework for connectome manipulations presented in this article, available under `https://github.com/BlueBrain/connectome-manipulator`, together with additional use case examples as summarized in Table S6.

parquet-converters: External dependency required for automatically converting the individual output .parquet files produced by Connectome-Manipulator to an output connectome in SONATA format, available under `https://github.com/BlueBrain/parquet-converters`.

SSCx network model: Seven column subvolume of a detailed model of the rat somatosensory cortex (Reimann et al., 2022; Isbister et al., 2023) in SONATA format used as a basis for all manipulations in this article, available under DOI 10.5281/zenodo.8026353.

SSCx flat coordinates: Coordinate mapping for the SSCx network model to a flat coordinate system aligned with the cortical layers, which has been released in Bolaños-Puchet et al. (2024) under 10.5281/zenodo.10686776.

SSCx connectome manipulation code: Repository with code and configuration files for applying manipulations to the SSCx network model, analyzing results, and reproducing the figures in this article; will be made available on `https://github.com` upon journal submission.

SSCx connectome manipulation data: Dataset containing the resulting data of the manipulated SSCx network model, such as fitted stochastic models, manipulated connectomes, structural validations, as well as simulation data; will be made available on `https://zenodo.org` upon journal submission.

Simulator software: Simulator software Neurodamus, which is a simulation control application for the NEURON simulator, released in Isbister et al. (2023) under 10.5281/zenodo.8075202 for simulating the SSCx network model.

Author contributions

- Conceptualization: C.P., M.W.R.
- Data curation: C.P.
- Formal analysis: C.P.
- Investigation: C.P.
- Methodology: C.P., J.B.I., K.K., M.W.R.
- Project administration: M.W.R.
- Software: C.P., O.A., K.K., M.W.
- Supervision: M.W.R.
- Validation: C.P., K.K.
- Visualization: C.P.
- Writing - original draft: C.P., M.W.R.
- Writing - review & editing: C.P., O.A., J.B.I., K.K., M.W., M.W.R.

References

Aru, J., Larkum, M. E., and Shine, J. M. (2023). The feasibility of artificial consciousness through the lens of neuroscience. *Trends in Neurosciences*, 46(12):1008–1017.

Billeh, Y. N. et al. (2020). Systematic integration of structural and functional data into multi-scale models of mouse primary visual cortex. *Neuron*, 106(3):388–403.

Bolaños-Puchet, S., Teska, A., Hernando, J. B., Lu, H., Romani, A., Schürmann, F., and Reimann, M. W. (2024). Enhancement of brain atlases with laminar coordinate systems: Flatmaps and barrel column annotations. *bioRxiv*.

Buzsáki, G. and Mizuseki, K. (2014). The log-dynamic brain: how skewed distributions affect network operations. *Nature Reviews Neuroscience*, 15(4):264–278.

Dai, K., Hernando, J., Billeh, Y. N., Gratiy, S. L., Planas, J., Davison, A. P., Dura-Bernal, S., Gleeson, P., Devresse, A., Dichter, B. K., et al. (2020). The sonata data format for efficient description of large-scale network models. *PLoS computational biology*, 16(2):e1007696.

De Kock, C., Bruno, R. M., Spors, H., and Sakmann, B. (2007). Layer-and cell-type-specific suprathreshold stimulus representation in rat primary somatosensory cortex. *The Journal of physiology*, 581(1):139–154.

Deco, G. and Hugues, E. (2012). Neural Network Mechanisms Underlying Stimulus Driven Variability Reduction. *PLoS Computational Biology*, 8(3):e1002395.

Douglas, R. J. and Martin, K. A. (2004). Neuronal circuits of the neocortex. *Annu. Rev. Neurosci.*, 27:419–451.

Dura-Bernal, S., Griffith, E. Y., et al. (2023). Data-driven multiscale model of macaque auditory thalamocortical circuits reproduces in vivo dynamics. *Cell Rep.*, 42(11).

Egas Santander, D., Pokorny, C., Ecker, A., Lazovskis, J., Santoro, M., Smith, J. P., Hess, K., Levi, R., and Reimann, M. W. (2024). Efficiency and reliability in biological neural network architectures. *bioRxiv*.

Gal, E. et al. (2017). Rich cell-type-specific network topology in neocortical microcircuitry. *Nat. Neurosci.*, 20(7):1004–1013.

Gal, E., Perin, R., Markram, H., London, M., and Segev, I. (2020). Neuron geometry underlies universal network features in cortical microcircuits. *bioRxiv*.

Gershman, S. J. (2024). What have we learned about artificial intelligence from studying the brain? *Biological Cybernetics*, pages 1–5.

Gopinath, N. (2023). Artificial intelligence and neuroscience: An update on fascinating relationships. *Process Biochemistry*, 125:113–120.

Isbister, J. B., Ecker, A., Pokorny, C., Bolaños-Puchet, S., Santander, D. E., et al. (2023). Modeling and simulation of neocortical micro- and mesocircuitry. Part II: Physiology and experimentation. *bioRxiv*.

Kumbhar, P., Hines, M., Fouriaux, J., Ovcharenko, A., King, J., Delalondre, F., and Schürmann, F. (2019). Coreneuron: an optimized compute engine for the neuron simulator. *Frontiers in neuroinformatics*, 13:63.

Lagzi, F. and Rotter, S. (2015). Dynamics of Competition between Subnetworks of Spiking Neuronal Networks in the Balanced State. *PLOS ONE*, 10(9):e0138947.

Litwin-Kumar, A. and Doiron, B. (2012). Slow dynamics and high variability in balanced cortical networks with clustered connections. *Nature Neuroscience*, 15(11):1498–1505.

Markram, H., Muller, E., Ramaswamy, S., Reimann, M. W., et al. (2015). Reconstruction and simulation of neocortical microcircuitry. *Cell*, 163(2):456–492.

MICrONS Consortium et al. (2021). Functional connectomics spanning multiple areas of mouse visual cortex. *bioRxiv*.

Miller, K. D. (2016). Canonical computations of cerebral cortex. *Current opinion in neurobiology*, 37:75–84.

Okun, M., Steinmetz, N. A., Cossell, L., Iacaruso, M. F., Ko, H., Barthó, P., Moore, T., Hofer, S. B., Mrsic-Flogel, T. D., Carandini, M., et al. (2015). Diverse coupling of neurons to populations in sensory cortex. *Nature*, 521(7553):511–515.

Olshausen, B. A. and Field, D. J. (2006). What is the other 85 percent of v1 doing. *L. van Hemmen, & T. Sejnowski (Eds.)*, 23:182–211.

Perin, R., Berger, T. K., and Markram, H. (2011). A synaptic organizing principle for cortical neuronal groups. *PNAS - Proceedings of the National Academy of Sciences*, 108(13):5419–5424. 387 citations (Crossref) [2021-09-14] Publisher: National Academy of Sciences.

Pfeffer, C. K., Xue, M., He, M., Huang, Z. J., and Scanziani, M. (2013). Inhibition of inhibition in visual cortex: the logic of connections between molecularly distinct interneurons. *Nature neuroscience*, 16(8):1068–1076.

Pi, H.-J., Hangya, B., Kvitsiani, D., Sanders, J. I., Huang, Z. J., and Kepecs, A. (2013). Cortical interneurons that specialize in disinhibitory control. *Nature*, 503(7477):521–524.

Reid, A. T., Headley, D. B., Mill, R. D., Sanchez-Romero, R., Uddin, L. Q., Marinazzo, D., Lurie, D. J., Valdés-Sosa, P. A., Hanson, S. J., Biswal, B. B., et al. (2019). Advancing functional connectivity research from association to causation. *Nature neuroscience*, 22(11):1751–1760.

Reimann, M. W., Bolaños-Puchet, S., Courcol, J.-D., Santander, D. E., et al. (2022). Modeling and simulation of neocortical micro- and mesocircuitry. Part I: Anatomy. *bioRxiv*.

Reimann, M. W., King, J. G., Muller, E. B., Ramaswamy, S., and Markram, H. (2015). An algorithm to predict the connectome of neural microcircuits. *Frontiers in computational neuroscience*, 9:120.

Renart, A., Moreno-Bote, R., Wang, X.-J., and Parga, N. (2007). Mean-Driven and Fluctuation-Driven Persistent Activity in Recurrent Networks. *Neural Computation*, 19(1):1–46.

Reyes-Puerta, V., Sun, J.-J., Kim, S., Kilb, W., and Luhmann, H. J. (2015). Laminar and columnar structure of sensory-evoked multineuronal spike sequences in adult rat barrel cortex in vivo. *Cerebral Cortex*, 25(8):2001–2021.

Rost, T., Deger, M., and Nawrot, M. P. (2018). Winnerless competition in clustered balanced networks: inhibitory assemblies do the trick. *Biological Cybernetics*, 112(1-2):81–98.

Rössert, C., Pozzorini, C., Chindemi, G., Davison, A. P., Eroe, C., King, J., Newton, T. H., Nolte, M., Ramaswamy, S., Reimann, M. W., Wybo, W., Gewaltig, M.-O., Gerstner, W., Markram, H., Segev, I., and Muller, E. (2017). Automated point-neuron simplification of data-driven microcircuit models.

Schneider-Mizell, C. M., Bodor, A., Brittain, D., Buchanan, J., Bumbarger, D. J., Elabbady, L., Kapner, D., Kinn, S., Mahalingam, G., Seshamani, S., Suckow, S., Takeno, M., Torres, R., Yin, W., Dorkenwald, S., Bae, J. A., Castro, M. A., Fahey, P. G., Froudakis, E., Halageri, A., Jia, Z., Jordan, C., Kemnitz, N., Lee, K., Li, K., Lu, R., Macrina, T., Mitchell, E., Mondal, S. S., Mu, S., Nehoran, B., Papadopoulos, S., Patel, S., Pitkow, X., Popovych, S., Silversmith, W., Sinz, F. H., Turner, N. L., Wong, W., Wu, J., Yu, S.-c., MICrONS Consortium, Reimer, J., Tolias, A. S., Seung, H. S., Reid, R. C., Collman, F., and Da Costa, N. M. (2023). Cell-type-specific inhibitory circuitry from a connectomic census of mouse visual cortex. preprint, Neuroscience.

Song, S., Sjöström, P. J., Reigl, M., Nelson, S., and Chklovskii, D. B. (2005). Highly Nonrandom Features of Synaptic Connectivity in Local Cortical Circuits. *PLoS Biology*, 3(3):e68.

Strata, P., Harvey, R., et al. (1999). Dale's principle. *Brain research bulletin*, 50(5):349–350.

Surianarayanan, C., Lawrence, J. J., Chelliah, P. R., Prakash, E., and Hewage, C. (2023). Convergence of artificial intelligence and neuroscience towards the diagnosis of neurological disorders—a scoping review. *Sensors*, 23(6):3062.

Verzelli, P., Tchumatchenko, T., and Kotaleski, J. H. (2024). Editorial overview: Computational neuroscience as a bridge between artificial intelligence, modeling and data. *Current Opinion in Neurobiology*, 84:102835.

Winding, M., Pedigo, B. D., Barnes, C. L., Patsolic, H. G., Park, Y., Kazimiers, T., Fushiki, A., Andrade, I. V., Khandelwal, A., Valdes-Aleman, J., et al. (2023). The connectome of an insect brain. *Science*, 379(6636):eadd9330.

Wohrer, A., Humphries, M. D., and Machens, C. K. (2013). Population-wide distributions of neural activity during perceptual decision-making. *Progress in neurobiology*, 103:156–193.

Supplementary material to

"A connectome manipulation framework for the systematic and reproducible study of structure–function relationships through simulations"

Christoph Pokorny[1], Omar Awile[1], James B. Isbister[1], Kerem Kurban[1], Matthias Wolf[1], and Michael W. Reimann[1]

[1]Blue Brain Project, École Polytechnique Fédérale de Lausanne (EPFL), Campus Biotech, 1202 Geneva, Switzerland

Supplementary figures

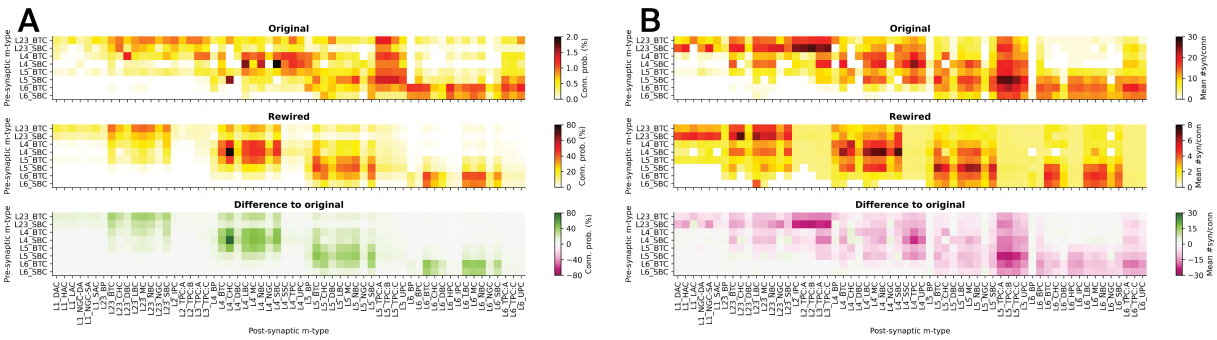

Figure S1: Detailed changes in rewired VIP+ interneuron connectivity by m-type. A: Mean connection probabilities in the original and rewired connectomes, and the differences between the rewired and original one. Note the different color scales. **B:** Same as A, but for the mean number of synapses per connection.

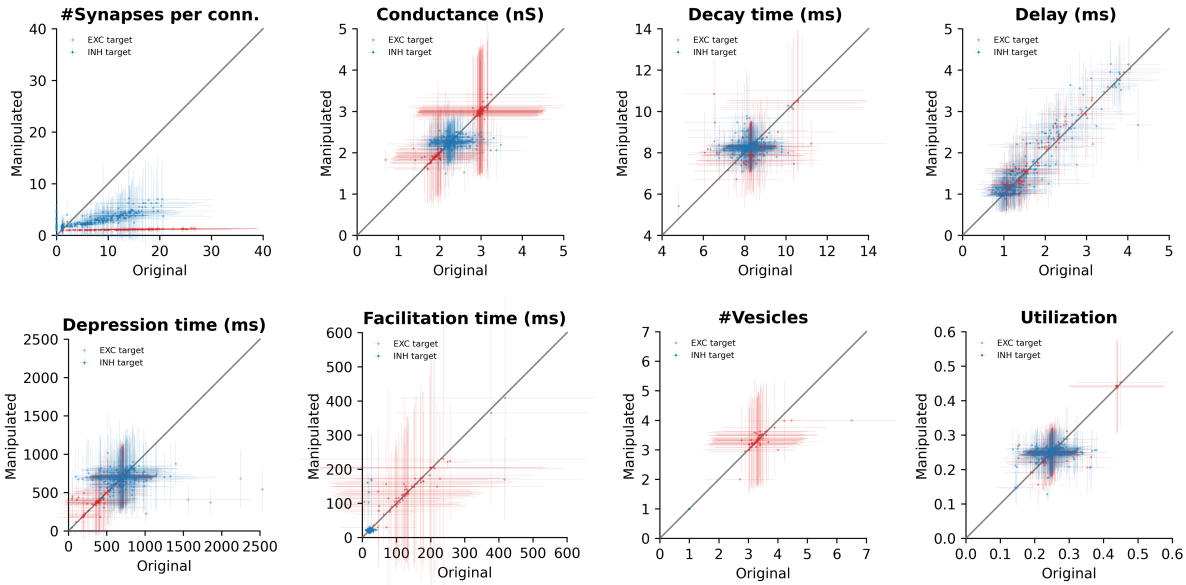

Figure S2: Validation of synapse parameters in rewired VIP+ interneuron connectivity by m-type. Numbers of synapses per connection and important physiological synapse parameter values in the original connectome (x axis) plotted against the manipulated connectome (y axis). Each data point is the mean ± SD of all connections belonging to a pathway, i.e., a pair of 8×60 pre- and post-synaptic m-types. Post-synaptic m-types are divided into excitatory (18; red) and inhibitory (42; blue) types, as indicated by the legend.

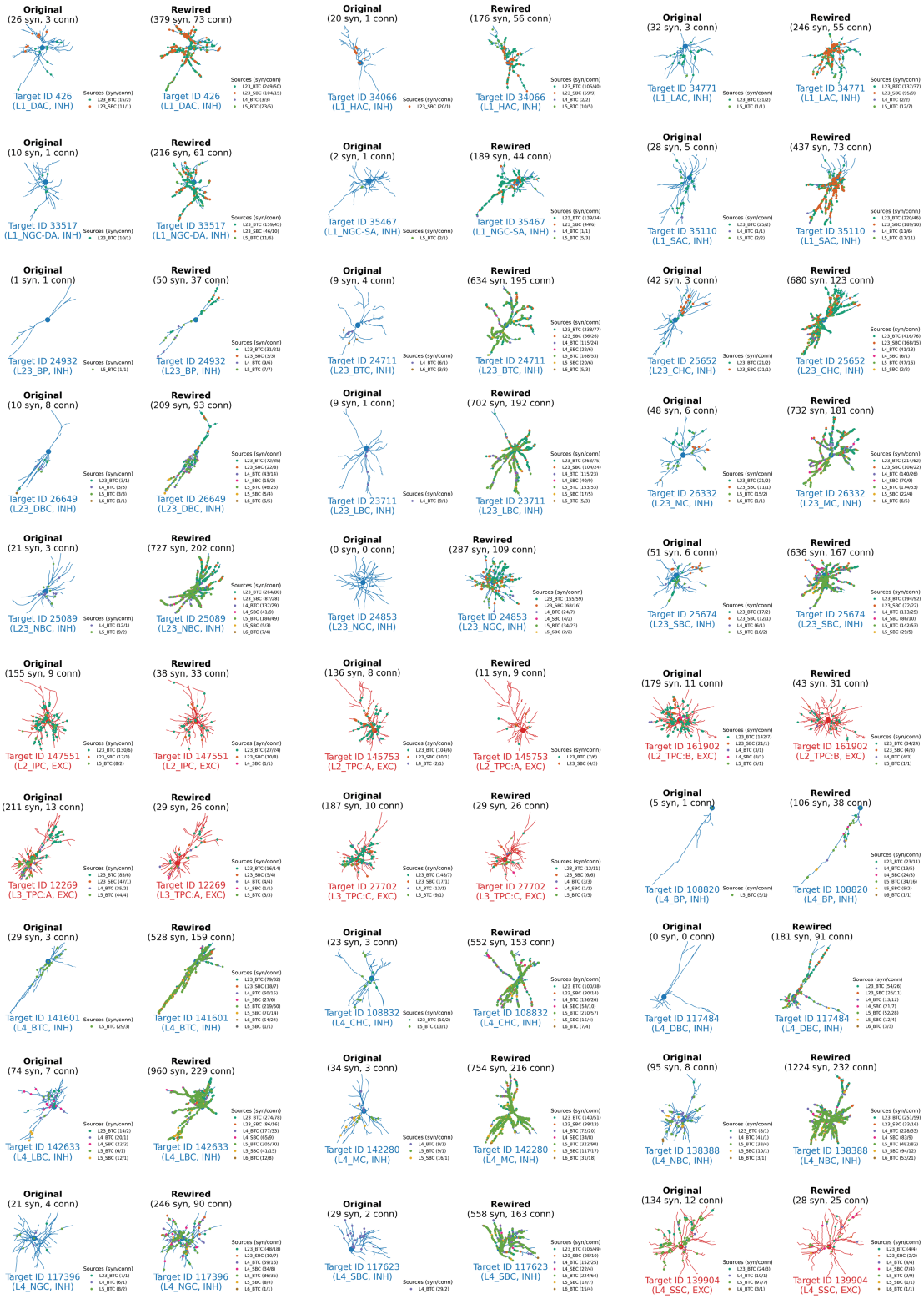

Figure S3: Examples of synapses on dentritic morphologies before and after VIP+ interneuron rewiring for 30 m-types. The dots indicate synapses from different BTC/SBC source types (as indicated by the legend) targeting inhibitory (blue) and excitatory (red) neurons of different m-types in the original vs. the rewired connectome. Small numbers denote numbers of synapses and connections respectively. Example neurons with highest in-degrees were selected. Same style as the examples in Figure 2D. Continued in Figure S4.

Figure S4: Examples of synapses on dentritic morphologies before and after VIP+ interneuron rewiring for another 30 m-types. The dots indicate synapses from different BTC/SBC source types (as indicated by the legend) targeting inhibitory (blue) and excitatory (red) neurons of different m-types in the original vs. the rewired connectome. Small numbers denote numbers of synapses and connections respectively. Example neurons with highest in-degrees were selected. Same style as the examples in Figure 2D. Continuation from Figure S3.

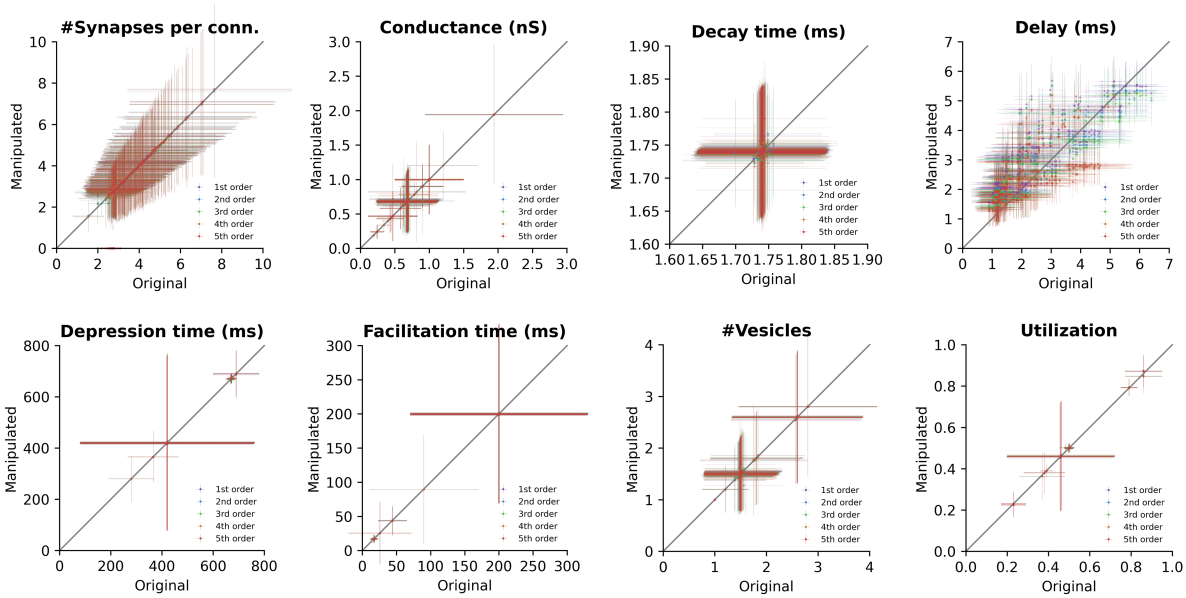

Figure S5: Validation of synapse parameters in the simplified connectomes by m-type. Numbers of synapses per connection and important physiological synapse parameter values in the original connectome (x axis) plotted against the simplified connectomes (y axis), as indicated by the legend. Each data point is the mean ± SD of all connections belonging to a pathway, i.e., a pair of 18×18 excitatory pre- and post-synaptic m-types (L2_IPC, L2_TPC:A, L2_TPC:B, L3_TPC:A, L3_TPC:C, L4_SSC, L4_TPC, L4_UPC, L5_TPC:A, L5_TPC:B, L5_TPC:C, L5_UPC, L6_BPC, L6_HPC, L6_IPC, L6_TPC:A, L6_TPC:C, L6_UPC).

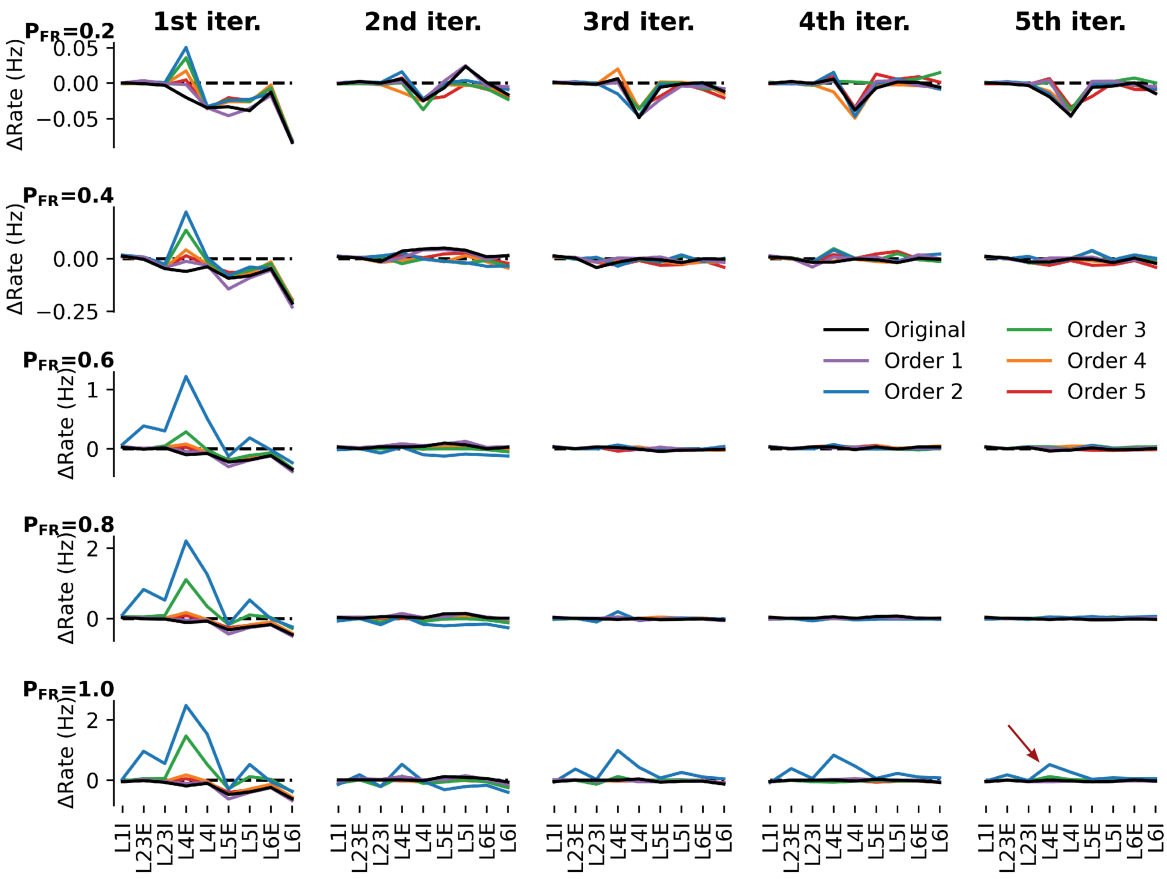

Figure S6: Spontaneous activity calibration of the simplified connectomes. We calibrated population-specific OU-parameters in order to obtain *in vivo*-like spontaneous activity (see Methods) for different fractions P_{FR} (rows) over five iterations (columns). Each individual plot show the firing rate differences ΔRate (y axis) between the observed rates and the fraction P_{FR} of the reference rates for different E/I populations (x axis). The individual colors denote different simplified connectomes as indicated by the legend. Note the different y axis scales. At $P_{FR} = 1.0$, there was still a rate mismatch for the 2nd order connectome even after five iterations (red arrow).

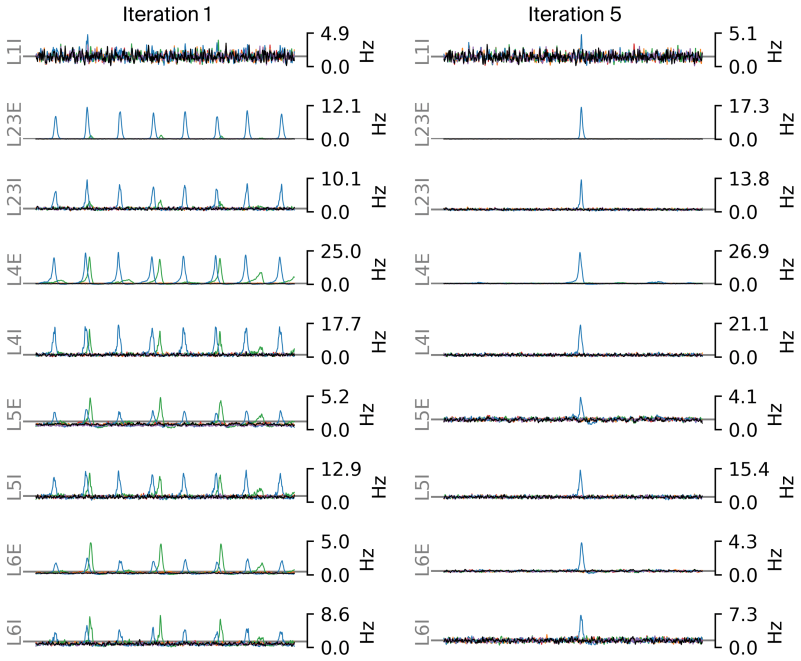

Figure S7: Network activity before and after re-calibration of the simplified connectomes for $P_{FR} = 1.0$. Instantaneous firing rates of 5 s spontaneous activity of E and I populations across layers for each of the rewired connectomes before and after re-calibration as in Figure 4, but for an *in vivo* fraction $P_{FR} = 1.0$.

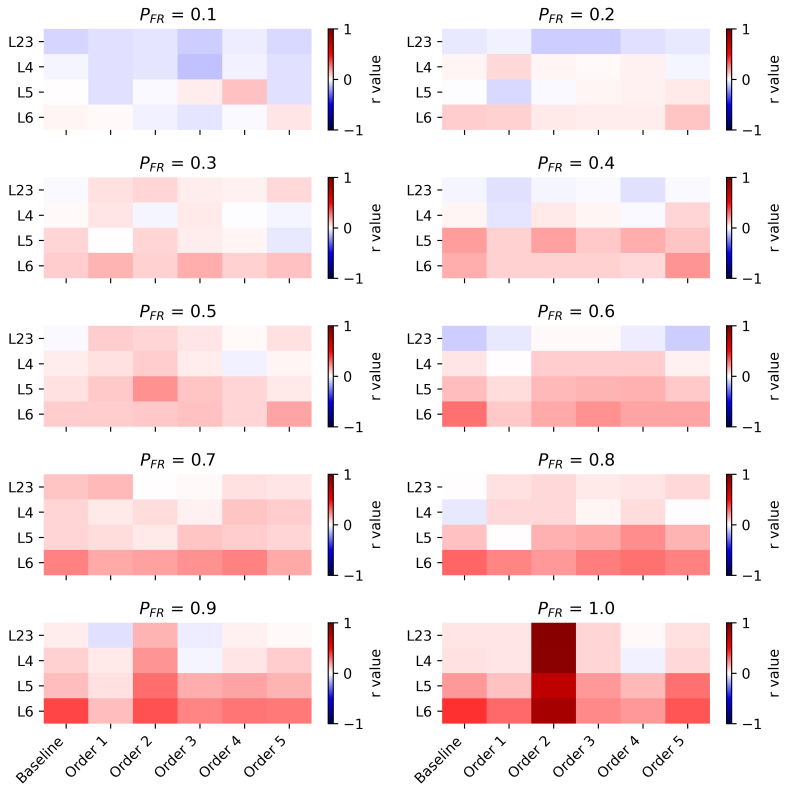

Figure S8: E/I correlation in the simplified connectomes after re-calibration. Pearson correlation coefficient (r value) between the instantaneous firing rates (5 ms bin size, Gaussian smoothing with SD 1.0) of 5 s spontaneous activity of the respective E and I populations across layers and different *in vivo* fractions P_{FR}.

Supplementary tables

Table S1: Available stochastic and generic model descriptions and extensions in the connectome manipulation framework. The table describes how the different model types are defined in terms of their inputs and outputs. Unless noted otherwise, the pre-synaptic neuron is always indexed with i and the post-synaptic neuron with j. The models are grouped by category.

Model & Description

Connectivity:

`ConnProb1stOrderModel`

1st order (Erdős-Rényi) connection probability model (Gal et al., 2020); returns a constant connection probability

$$P_{ij} = p_{const} \tag{5}$$

between neurons i and j.

`ConnProb2ndOrderExpModel`

2nd order (distance-dependent) connection probability model (Gal et al., 2020), based on an exponential function; returns the connection probability

$$P_{ij} = f(d_{ij}) = \alpha \cdot e^{-\beta \cdot d_{ij}} \tag{6}$$

which is a function of the Euclidean (intersomatic) distance d_{ij} between neurons i and j.

`ConnProb2ndOrderComplexExpModel`

2nd order (distance-dependent) connection probability model (Gal et al., 2020), based on a proximal (fast-decaying) and a distal (slowly-decaying) exponential function; returns the connection probability

$$P_{ij} = f(d_{ij}) = \alpha_p \cdot e^{-\beta_p \cdot {d_{ij}}^{\gamma}} + \alpha_d \cdot e^{-\beta_d \cdot d_{ij}} \tag{7}$$

which is a function of the Euclidean (intersomatic) distance d_{ij} between neurons i and j.

`ConnProb3rdOrderExpModel`

3rd order (bipolar distance-dependent) connection probability model (Gal et al., 2020), based on an exponential function; returns the connection probability

$$P_{ij} = f(d_{ij}, sign(\Delta z)) = \begin{cases} \alpha_- \cdot e^{-\beta_- \cdot d_{ij}}, \text{if } sign(\Delta z) < 0 \\ \alpha_+ \cdot e^{-\beta_+ \cdot d_{ij}}, \text{if } sign(\Delta z) > 0 \\ \text{Mean of both}, \text{if } sign(\Delta z) = 0 \end{cases} \tag{8}$$

which is a function of the Euclidean (intersomatic) distance d_{ij} between neurons i and j and the sign of the difference of their vertical positions (i.e., along z axis; $\Delta z = z_j - z_i$).

`ConnProb3rdOrderComplexExpModel`

3rd order (bipolar distance-dependent) connection probability model (Gal et al., 2020), based on a proximal (fast-decaying) and a distal (slowly-decaying) exponential function; returns the connection probability

$$P_{ij} = f(d_{ij}, sign(\Delta z)) = \begin{cases} \alpha_{p-} \cdot e^{-\beta_{p-} \cdot {d_{ij}}^{\gamma_-}} + \alpha_{d-} \cdot e^{-\beta_{d-} \cdot d_{ij}}, \text{if } sign(\Delta z) < 0 \\ \alpha_{p+} \cdot e^{-\beta_{p+} \cdot {d_{ij}}^{\gamma_+}} + \alpha_{d+} \cdot e^{-\beta_{d+} \cdot d_{ij}}, \text{if } sign(\Delta z) > 0 \\ \text{Mean of both}, \text{if } sign(\Delta z) = 0 \end{cases} \tag{9}$$

which is a function of the Euclidean (intersomatic) distance d_{ij} between neurons i and j and the sign of the difference of their vertical positions (i.e., along z axis; $\Delta z = z_j - z_i$).

`ConnProb4thOrderLinInterpnModel`

4th order (offset-dependent) connection probability model (Gal et al., 2020), based on multidimensional linear interpolation of probability values given on a regular grid of x, y, and z offsets; returns the connection probability

$$P_{ij} = f(\Delta x, \Delta y, \Delta z) \tag{10}$$

which is a function of the x, y, and z offsets between neurons i and j (i.e., $\Delta x = x_j - x_i$, $\Delta y = y_j - y_i$, $\Delta z = z_j - z_i$).

`ConnProb4thOrderLinInterpnReducedModel`

4th order (offset-dependent) connection probability model (Gal et al., 2020), based on multidimensional linear interpolation of probability values given on a regular grid of radial (r; within x/y plane) and axial (z) offsets; returns the connection probability

$$P_{ij} = f(\Delta r, \Delta z) \tag{11}$$

which is a function of the offsets in radial and axial direction between neurons i and j (i.e., $\Delta r = \sqrt{\Delta x^2 + \Delta y^2}$, $\Delta z = z_j - z_i$).

`ConnProb5thOrderLinInterpnModel`

5th order (position-dependent) connection probability model (Gal et al., 2020), based on multidimensional linear interpolation of probability values given on a regular grid of x, y, and z positions and offsets; returns the connection probability

$$P_{ij} = f(x_i, y_i, z_i, \Delta x, \Delta y, \Delta z) \tag{12}$$

which is a function of the x, y, and z positions of the pre-synaptic neuron i and offsets between neurons i and j (i.e., $\Delta x = x_j - x_i$, $\Delta y = y_j - y_i$, $\Delta z = z_j - z_i$).

`ConnProb5thOrderLinInterpnReducedModel`

5th order (position-dependent) connection probability model (Gal et al., 2020), based on multidimensional linear interpolation of probability values given on a regular grid of axial (z) positions and radial (r; within x/y plane) and axial offsets; returns the connection probability

$$P_{ij} = f(z_i, \Delta r, \Delta z) \tag{13}$$

which is a function of the axial position of the pre-synaptic neuron i and the offsets in radial and axial direction between neurons i and j (i.e., $\Delta r = \sqrt{\Delta x^2 + \Delta y^2}$, $\Delta z = z_j - z_i$).

`ConnProbAdjModel`

Connection probability model, defined by a sparse adjacency matrix; returns the connection probability 1 if a connection between neurons i and j exists, otherwise 0, i.e., this is essentially a deterministic connectivity representation.

Physiology:

`LinDelayModel`

Linear distance-dependent axonal delay model, optionally with pathway-specific model attributes for different combinations of pre-/post-synaptic m-types; returns randomly drawn delay values from a truncated normal distribution with given mean

$$\mu_{ij} = \alpha_D + \beta_D * d_{ij}, \tag{14}$$

standard deviation σ_D (constant), and minimum delay D_{min} (constant), i.e., the mean delay μ_{ij} linearly depends on the Euclidean distance d_{ij} between the soma of a pre-synaptic neuron i and the synapse position on the post-synaptic dendrite of neuron j.

`ConnPropsModel`

Connection properties model for realizing connections by forming synapses, based on individual parameters for pairs of pre-/post-synaptic m-types; returns synapse properties (e.g., conductance, time constants, readily-releasable vesicles, etc.) drawn from given distributions, such as constant, (truncated) normal, gamma, (zero-truncated) Poisson, or discrete distributions, for a single connection. The number of synapses per connection are either drawn from a given distribution or provided externally. The same property values can be shared among all synapses belonging to the same connection; otherwise, they are drawn independently.

Model extensions:

`PosMapModel`

Position mapping from one coordinate system to another for a given set of neurons; returns the mapped soma position for a given neuron ID. This model extension can be used in combination with all connection probability models depending on spatial geometry, e.g., for aligning the coordinate axes with the cortical layers using a flat/depth map projection.
Importantly, when using a position mapping for fitting a connection probability model, the same position mapping must be used used when applying the connection probability model, e.g., for (re)wiring.

Generic:

`LookupTableModel`

Generic lookup table to access any (sparse) information based on pre- and post-synaptic neuron IDs; returns zero if no value is stored for a given pair of neurons. For example, it can be used to specify adjacency matrices (boolean entries), connection probabilities between neurons (decimal numbers between 0.0 and 1.0), or synaptome matrices (integer numbers of synapses per connection).

`PropsTableModel`

Generic properties table to store a table of properties based on pre- and post-synaptic neuron IDs; it may contain multiple entries for a given pair of neurons. For example, it can be used to store exact synapse positions precomputed externally.

Table S2: Model building tools in the connectome manipulation framework. The table describes the available tools (Python modules) for extracting data, fitting stochastic models, and storing them in a model representation format. All tools work in three steps: data extraction, model fitting (where applicable) and storing, and visualization of the actual data and/or model output.

Model & Description

Connectivity:

`conn_prob.py`

Model fitting of stochastic connection probability models with given model order (`ConnProb...Order` models in Table S1), based on binned connection probabilities (with given bin sizes) extracted from an existing connectome.

`conn_prob_adj.py`

Adjacency matrix extraction from an existing connectome and storing it as `ConnProbAdjModel` (Table S1).

Physiology:

`delay.py`

Axon delay model fitting of type `LinDelayModel` (Table S1) based on binned distance-dependent synaptic delays (with given bin size) between samples of neurons extracted from an existing connectome. Least squares linear regression is used to determine the model coefficient α_d and β_d for the mean delay (Eq. 14); its standard deviation σ_D is determined as the mean of the standard deviations of all bins, D_{min} as the minimum overall delay.

`conn_props.py`

Connection and synapse properties model fitting of type `ConnPropsModel` (Table S1) for pairs of pre-/post-synaptic m-types. Statistics for connections (in case of shared values among synapses) or individual synapses between samples of neurons are extracted from an existing connectome for each pair of m-types and used to parameterize the individual model distribution types specified for each property.

Missing values for pairs of m-types (i.e., between which there are no or too few connections to estimate reliable statistics) are gradually interpolated at different levels of granularity from similar pathways with matching pre-/post-synaptic m-type, layer, or E/I cell type, if available:

Level	Pre-synaptic	Post-synaptic
0	m-type	layer & cell type
1	m-type	cell type
2	layer & cell type	layer & cell type
3	cell type	cell type
4	any	any

Model extensions:

`pos_mapping.py`

Building a position mapping extension of type `PosMapModel` (Table S1) from a voxelized brain atlas in NRRD (Nearly Raw Raster Data) format consisting of a flat map (x/y plane, parallel to cortical layers) and a cortical depth map (z axis, perpendicular to cortical layer). The soma positions of a given population are transformed to the new coordinate system by linear interpolation between voxel positions, if possible; otherwise, nearest-neighbor interpolation is employed.

`pos_mapping_from_table.py`

Building a position mapping extension of type `PosMapModel` (Table S1), by just loading the transformed soma positions for a given population from a position table precomputed externally.

Table S3: Manipulation operations in the connectome manipulation framework. The table describes the available manipulation operations (Python modules) and how they can be configured. The modules are grouped by types of use cases as in Figure 1C.

Module & Description

Testing:

`null_manipulation.py`

Dummy manipulation not performing any manipulation at all, i.e., the output is identical to the input connectome. This operation is intended as a test condition to run the manipulation pipeline without actually manipulating the connectome.

Remove synapses:

`syn_removal.py`

Removes a certain percentage of randomly sampled synapses between selected populations of neurons. Optionally, connections can be kept (i.e., keeping at least 1 synapse per connection), and rescaling of synaptic conductances is possible in order to keep the sum of conductances per connection constant.

`syn_subsampling.py`

Random subsampling of synapses, keeping a certain percentage of randomly selected synapses. This is a special case of the functionality implemented in `syn_removal.py`, which is more efficient but does not have any selection options supported.

Alter synapses:

`syn_prop_alteration.py`

Alters values of selected synapse properties (e.g., conductance, delay, ...) of a certain percentage of randomly sampled synapses between selected populations of neurons. Values can be altered according to the following options:

Option	New values given by
`setval`	Absolute value
`scale`	Relative scaling by a given value
`offset`	Offsetting by a give value
`shuffle`	Shuffling among synapses
`randval`	Random absolute value drawn from a given distribution
`randscale`	Relative scaling by a value drawn from a given distribution
`randadd`	Adding a random value drawn from a given distribution

This operation can be utilized in various applications for investigating the impact of certain synapse properties (e.g., lower/higher conductances, shorter/longer delays, etc.) on the emerging network activity. It can also be used to (carefully) manipulate synapse locations on the dendrite, e.g., placing them on the soma, by setting their section index and offset to 0.

Remove connections:

`conn_removal.py`

Removes a certain percentage of randomly sampled connections (i.e., all synapses belonging to a connection) between selected populations of neurons. Optionally, only connections within a certain range of synapses per connection, and/or within a specific connection mask can be removed.

A potential application of this manipulation is to systematically study the impact of missing connections on the network activity, as in lesion experiments.

`conn_extraction.py`

Extracts the connectome of a given node set (i.e., a named sub-population of neurons), keeping only connections between neurons within that node set and removing all connections from, to, and between neurons outside that node set. Node sets intrinsic to a circuit or provided through an external SONATA node sets file (JSON format) containing a list of neuron IDs are supported. Potential applications are restricting the connectivity to certain regions of interest, or as pre-processing step to reduce the connectome size in order to obtain a new baseline connectome for further manipulations.

Rewire, transplant, wire:

`conn_rewiring.py`

Fundamental operation for (re)wiring connections between populations of neurons, which involves removing and changing existing, as well as creating new connections. Connectivity is defined by a given connection probability model (`ConnProb...` models in Table S1) with or without use of a position mapping (`PosMapModel`), which can optionally be scaled by a global probability scaling factor `p_scale` to fine-tune the resulting number of connections. Existing connections can be preserved or reused in new connections; existing synapses (and connections) may be deleted and new ones created by either reusing existing or randomly generating new positions on the post-synaptic dendrites, or loading existing positions externally. Also, in-degrees can be preserved, and rewiring can be restricted to only adding or deleting connections; see Table S4 for all options. Importantly, (re)wiring operations are restricted to outgoing connections from either excitatory or inhibitory neuron types at a time, to prevent intermixing of different synapse classes when reusing synapses in new connections, which would be a violation of Dale's law (Strata et al., 1999).

Physiological synapse properties can be sampled from existing synapses or drawn from pathway-specific model distributions (`ConnPropsModel` in Table S1); synaptic delays can be assigned depending on the distance to their pre-synaptic neurons (`LinDelayModel`). Numbers of synapses per connection can be sampled from existing connections, drawn from model distributions, or provided by a synaptome matrix (stored as `LookupTableModel`).

The main applications of this operation are to create connectivity for network models without connectome, and to investigate the impact of rewired or transplanted connectivity in existing connectomes.

Table S4: Rewiring options. The table describes the available options for rewiring (i.e., by use of `conn_rewiring`) and how they affect the resulting connectivity.

Option & Description
Connections:
`keep_conns`
Keeping existing connections. If during the connectivity assignment step (see Figure 1D1) connections are to be established that already exist in the input connectome, such connections are kept exactly as they are. That is, the pre-synaptic neuron (i.e., connection source), number of synapses per connection, synapse positions on the post-synaptic dendrite, as well as synapse physiology are preserved.
`reuse_conns`
Reusing existing connections. During rewiring, existing connections are reused to realize new connections in the rewired connectome. Specifically, synapses per connection, synapse positions on the post-synaptic dendrite as well as synapse physiology are preserved, but new pre-synaptic neurons are randomly assigned to these connections. If the number of incoming connections to be established with a post-synaptic neuron exceeds the number of existing (incoming) connections, all existing ones are reused and the remaining ones created as in the general rewiring case. If the number of existing connections is higher, a subset of them are randomly selected and reused, and the remaining ones are deleted.
`keep_conns & reuse_conns`
In case both options are selected, the behavior will be like `keep_conns` when establishing connections that already exist in the input connectome. For connections that exist in the input but no longer in the rewired connectome, the behavior is like `reuse_conns`, i.e., such connections may be reused to establish other new (incoming) connections.
Otherwise
Disregarding existing connections. In the general rewiring case, existing connections are neither kept nor reused but replaced by newly generated (incoming) connections.
Synapse physiology:
`sample`
Generation method for physiological parameterization based on sampling. When creating new synapses to form new connections, the number of synapses per connection as well as their physiological property values are randomly sampled from other existing synapses. The existing synapses to sample from are always of the same synapse class as the new synapses that are to be created (i.e., either excitatory or inhibitory). Also, only synapses on post-synaptic neurons belonging to the same m-type as the post-synaptic target neuron are considered. If such synapses don't exist, synapses of the same layer (but non-matching m-type) are considered. If such synapses don't exist either, synaptic properties are sampled from synapses of all available post-synaptic neurons (i.e., neither matching m-type nor layer). When rewiring is run with more than one data split (see Figure 5), property values can only be sampled from synapses within the same split, in which case the `randomization` method is highly recommended which is independent of data splits.
`randomize`
Generation method for physiological parameterization based on randomization. When creating new synapses to form new connections, their physiological property values are randomly drawn from given (pathway-specific) property distributions (Table S1). Numbers of synapses per connection are either randomly drawn as well or can be provided through a synaptome matrix.

Synapse positions:

`reuse`

Reusing existing synapse positions. Existing synapse positions on the post-synaptic dendrite are reused when generating new synapses. Specifically, for each new connections to be created, the corresponding number of synapses are randomly sampled from all existing synapses, and new synapses are placed at their exact positions. For multi-synaptic connections, the synapse positions are drawn without replacement, if possible. The advantage of reusing positions is that their overall distribution of positions is preserved (but not for a single connection), and that rewiring runs faster since no access to dendritic morphologies is required. A potential drawback is that multiple synapses belonging to the same or different connections may be placed at exactly the same position, especially for post-synaptic neurons with few existing synapses.

`randomize`

Randomizing synapse positions. New positions are randomly drawn based on the actual dendritic morphologies. Specifically, each new position is uniformly drawn from the soma and all dendritic morphology sections, and a random (relative) offset within each section (except soma, which has zero offset by definition). This method has lower risk of duplicate synapse positions, but is slower since access to dendritic morphologies and recomputation of 3D synapse positions based on their drawn sections and offsets are required.

`external`

Loading external synapse positions. New positions are directly loaded from an external position table (provided as `PropsTableModel`, see Table S1). An error is raised if not enough positions are available for a given connection. In case the table contains more positions than required, they are sequentially loaded. This option should be used with caution, since external positions are loaded without consistency checks against actual morphologies.

In-degree:

`keep_indegree`

Keeping the in-degrees constant. The number of incoming connections for each post-synaptic neuron which is subject to rewiring is preserved. This is achieved in the connectivity assignment step (see Figure 1D1), by drawing exactly the same number of pre-synaptic neurons as in the input connectome to be connected with a post-synaptic neuron.

This imposes certain constraints on the rewiring operation: (i) The connection probabilities obtained from a stochastic connection probability model (see Table S1) are not interpreted in absolute but in relative terms, so that the overall distribution of connection probabilities may not be fulfilled exactly, but only for each post-synaptic neuron independently. (ii) At least the required number of pre-synaptic neurons must have non-zero connection probabilities to be connected with a post-synaptic neuron, otherwise keeping the in-degree is not possible. (iii) Keeping the in-degree is not compatible with rewiring options that allow only adding or only deleting connections.

Otherwise

Disregarding existing in-degrees. The numbers of incoming connections for each post-synaptic neuron are not guaranteed to be preserved. Instead, any number of pre-synaptic neurons may be assigned to be connected with a post-synaptic neuron based on the connection probabilities obtained from a stochastic connection probability model (see Table S1).

Restricted rewiring:

`add_only`

During connectivity assignment (see Figure 1D1), connections can be only added; no existing connections will be deleted (but may be re-parameterized depending on the other options selected). Hence, the resulting number of incoming connections to a post-synaptic neuron is always greater than or equal to the respective number in the input connectome.

`delete_only`

During connectivity assignment (see Figure 1D1), connections can be only deleted; no new connections will be established. Hence, the resulting number of incoming connections to a post-synaptic neuron is always less than or equal to the respective number in the input connectome.

Otherwise

No restrictions on rewiring; that is, new connections may be established and existing ones deleted.

Matching total number of connections (see Section 4.1.4 in Methods):

`p_scale`

Global scaling factor for adjusting the connection probabilities given by a connectivity model (see Table S1).

`estimation_run`

Rewiring operation with early stopping, which does not generate an actual connectome or output file, but writes an estimate of the average number of incoming connections for each post-synaptic neuron into a data log file.

`opt_nconn`

The drawn number of incoming connections during connectivity assignment (see Figure 1D1) in a single random instance of the connectome will be optimized to match its expected number of connections on average.

Table S5: Tools for structural comparison in the connectome manipulation framework. The table describes the available tools (Python modules) for comparing connectomes in terms of connectivity structure and synaptic physiology. All tools work in two steps: computing metrics to compare, and visualizing them for the two connectomes and their difference.

Module & Description

Connectivity:

`adjacency.py`

Structural comparison of two connectomes in terms of adjacency and synaptome matrices (i.e., connectivity and synapse counts between pairs of individual pre-/post-synaptic neurons) for selected pathways.

`connectivity.py`

Structural comparison of two connectomes in terms of connection probability and mean number of synapses per connection between groups of neurons (grouped by a given cell property, e.g., by layer, m-types, etc.) for selected pathways.

Physiology:

`properties.py`

Comparison of two connectomes in terms of statistical properties (e.g., mean, standard deviation, etc.) of selected synapse parameters of connections between groups of neurons (grouped by a given cell property, e.g., by layer, m-types, etc.) for selected pathways.

Table S6: Overview of Jupyter notebooks to demonstrate different use case examples. The following Jupyter notebooks are contained in the code repository `https://github.com/BlueBrain/connectome-manipulator` in the `/examples` subfolder. Step-by-step instructions are contained therein.

Example & Description

Basic examples:

`null_manipulation`

Minimal working example to test the connectome manipulation pipeline without performing any actual manipulation, i.e., the output connectome will be identical to the input connectome.