

PAPER • OPEN ACCESS

## Avoidance, adjacency, and association in distributed systems design

To cite this article: Andrei A Klishin *et al* 2021 *J. Phys. Complex.* **2** 025015

View the [article online](#) for updates and enhancements.

## OPEN ACCESS

## PAPER



## Avoidance, adjacency, and association in distributed systems design

## RECEIVED

10 October 2020

## REVISED

21 January 2021

## ACCEPTED FOR PUBLICATION

2 February 2021

## PUBLISHED

8 April 2021

Original content from this work may be used under the terms of the [Creative Commons Attribution 4.0 licence](https://creativecommons.org/licenses/by/4.0/).

Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

Andrei A Klishin<sup>1,2,3,6</sup> , David J Singer<sup>4</sup> and Greg van Anders<sup>1,2,5,\*</sup> <sup>1</sup> Department of Physics, University of Michigan, Ann Arbor, Michigan 48109, United States of America<sup>2</sup> Center for the Study of Complex Systems, University of Michigan, Ann Arbor, Michigan 48109, United States of America<sup>3</sup> John A Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138, United States of America<sup>4</sup> Naval Architecture and Marine Engineering, University of Michigan, Ann Arbor, Michigan 48109, United States of America<sup>5</sup> Department of Physics, Engineering Physics, and Astronomy, Queen's University, Kingston, Ontario, K7L 3N6, Canada<sup>6</sup> Current address: Department of Bioengineering, School of Engineering & Applied Science, University of Pennsylvania, Philadelphia, PA 19104, United States of America.

\* Author to whom any correspondence should be addressed.

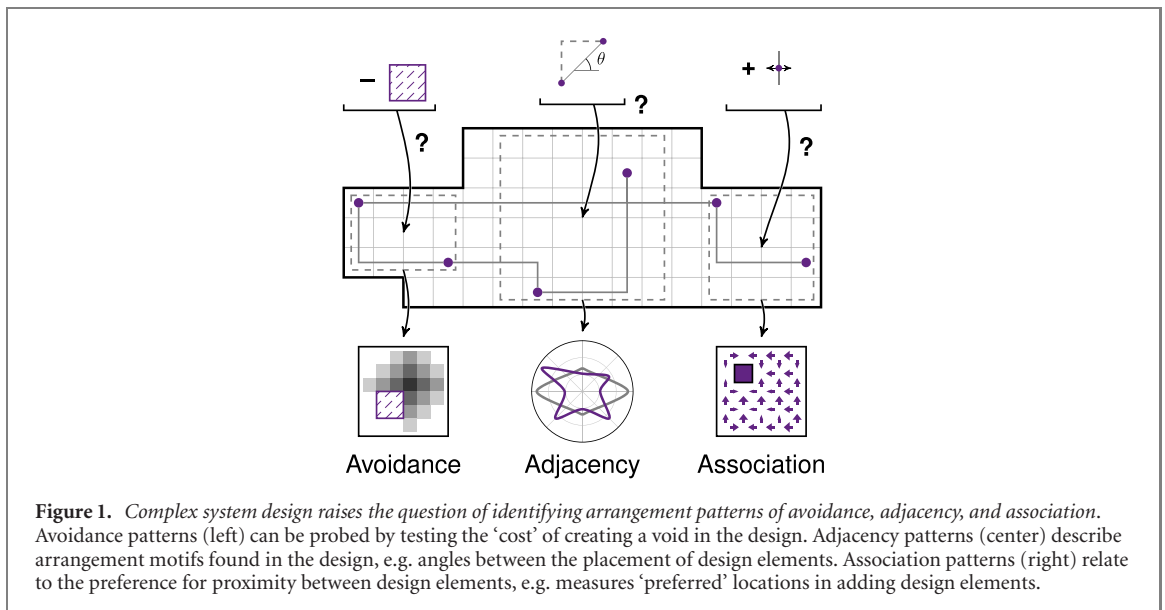
E-mail: [gva@queensu.ca](mailto:gva@queensu.ca)**Keywords:** statistical mechanics, systems physics, distributed systems design, logical architecture, physical architecture, naval architecture**Abstract**

Patterns of avoidance, adjacency, and association in complex systems design emerge from the system's underlying logical architecture (functional relationships among components) and physical architecture (component physical properties and spatial location). Understanding the physical–logical architecture interplay that gives rise to patterns of arrangement requires a quantitative approach that bridges both descriptions. Here, we show that statistical physics reveals patterns of avoidance, adjacency, and association across sets of complex, distributed system design solutions. Using an example arrangement problem and tensor network methods, we identify several phenomena in complex systems design, including placement symmetry breaking, propagating correlation, and emergent localization. Our approach generalizes straightforwardly to a broad range of complex systems design settings where it can provide a platform for investigating basic design phenomena.

A fundamental question in the design of complex, multicomponent systems is how the components of the system are arranged [1–3]. Arrangement problems, generically, present the challenge of anticipating or identifying ‘prime real estate’ [4, 5], i.e. sectors of the system's architecture that have premium or priority because of the mutual avoidance, adjacency, or association between system components (see figure 1). Determining or anticipating components' patterns of avoidance, adjacency, and association is important in so-called ‘greenfield’ settings, i.e. before design aspects have been specified, and in ‘brownfield’ settings, i.e. when one or more system design aspects have been determined [6–8]. In both greenfield and brownfield settings, determining the patterns of arrangement and identifying system factors driving those behaviors is crucial for managing, mitigating, or adapting to likely design outcomes [9].

Managing likely design outcomes by identifying patterns of arrangement depends crucially on both a system's logical architecture, i.e. the set of functional connections between components, and on the system's physical architecture, i.e. the physical properties of the components and their arrangement in space [10]. A system's logical architecture is essentially topological, and can be treated using network theory techniques [11]. In contrast, describing the physical architecture of a system is typically done by disciplinary engineering approaches that rest on known physical principles. Treating problems in arrangements that arise from an interplay of a system's logical and physical architecture requires a framework that bridges a system's network-theory-level description and its physical/spatial description. Whereas approaches exist at the network theory and at the physical spatial levels, how they can be bridged is an open question.

Here, we show that topological and physical descriptions of complex systems design can be bridged using statistical physics. Using statistical physics we demonstrate a framework that reveals patterns of avoidance, adjacency, and association in arrangement problems. We use an example arrangement problem to concretely



demonstrate how our framework can identify patterns of arrangement and how those patterns are driven by the design’s logical and physical architecture, in both greenfield and brownfield settings.

## 1. Systems physics framework

### 1.1. Motivation: design challenges lurk between logical and physical architectures

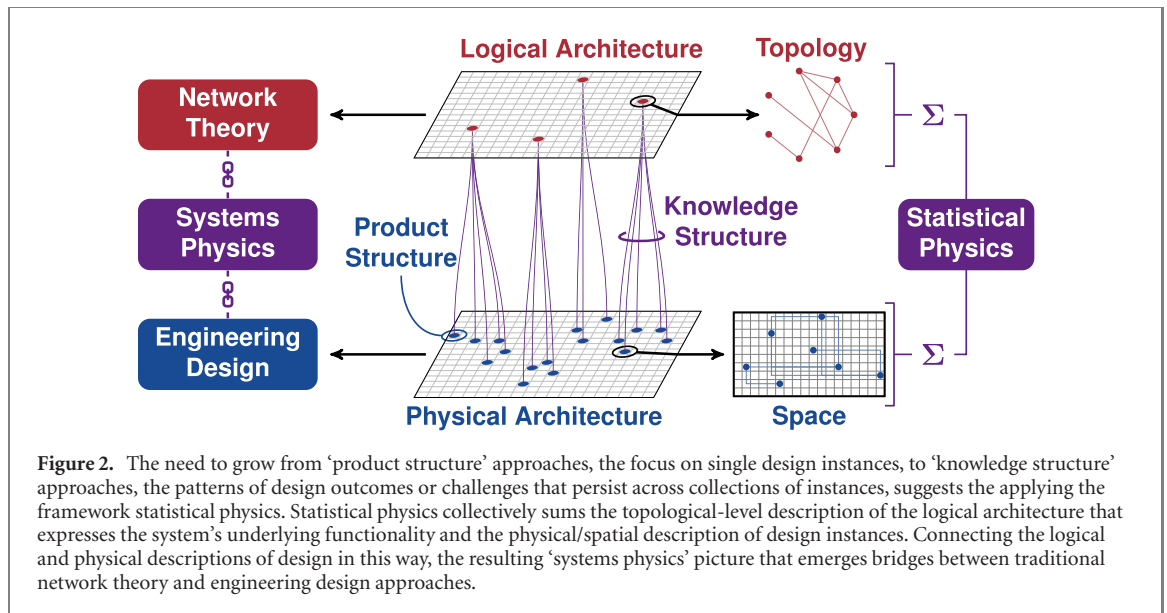
Complex systems are typically comprised by several interacting entities. The interactions among the entities are often described at two different levels: the description of what-is-connected-to-what, which is mathematically a graph-theoretic description, and by the description of how the entities physically interact with one another in space, which is described by physics. Taken separately, both levels of description give useful but incomplete insights into design.

The logical architecture’s graph-theoretic description of a complex system design is valuable because it isolates the connections between system elements that underly functionality [10]. Functionality in the logical architecture is reduced to the topology of connections, and this connection topology can be analyzed with network theory techniques [12–14]. Network theory approaches to analyzing logical architecture are powerful because they abstract out the system’s physical realization [15]. However, realizing the logical architecture physically can produce emergent functional connections that are lost when logical architecture is analyzed alone.

The physical architecture describes the realization of a complex system design in terms of physical entities with physical properties. Whereas entities in the logical architecture have abstract interactions that are encoded topologically, in the physical architecture interactions interact mechanically, thermodynamically, electromagnetically, etc, depending on physical factors such as energy consumption and proximity in space. By retaining this level of detail, the physical architecture provides an intimate picture of the performance of design elements. However, this intimate portrait of performance typically describes a single physical architecture instance. What that single instance means for the space of possible designs more generally is often unclear.

Though they can provide key insight into single design instances, both physical and logical architecture descriptions restrict our ability to understand general characteristics of design. This restriction exists because *general design characteristics are properties of design problem spaces rather than of design instances*. The focus on design instances has been described previously as design organized around ‘product structures’, i.e., around a particular outcome of the design process [16]. Contrasting with product structures are ‘knowledge structures’ that organize the design process around relationships between design elements that persist across instances [17].

Searching across instances is key for identifying patterns of avoidance, adjacency, and association that are generic features of design problem spaces. Achieving this requires a different approach. To formulate this approach, the key challenge is in framing knowledge that emerges from collections of possible instances of a system. The problem of many instances that give rise to collective behavior is the underlying principle that motivated the development of statistical mechanics [18]. The fact that an analogous problem emerges in design,



i.e. the need to formulate knowledge structures to identify patterns in design space, suggests that statistical physics could serve as the foundation for a similar approach. Figure 2 illustrates this strategy of attack.

## 1.2. Statistical physics approach

The need to address the problem of identifying patterns of avoidance, adjacency, and association that persist across spaces of designs points to statistical physics as a framework. To construct this framework there are two key challenges: formulating the design problem as a statistical mechanics model, and extracting from the model the knowledge structures that encode design space properties.

To construct statistical physics models of design, we need two things: the space of states and some metric on this space. For design problems that are studied with optimization techniques like simulated annealing, these two things are already at hand. A generic approach for constructing a statistical physics design framework given a space of possible designs and a set of design objectives was developed in reference [19].

*Mathematical Formulation*—Following reference [19], we denote the *design space* as the set  $\{\alpha\}$  consisting of individual design solutions  $\alpha$ . Each solution  $\alpha$  can be quantitatively evaluated with a *design objective*  $\mathcal{O}(\alpha)$ . Instead of exclusively focusing on the design that minimizes  $\mathcal{O}$ , we consider a probability distribution  $p_\alpha$  over designs [19]. Of all possible normalized probability distributions, we seek one that maximizes the Shannon entropy functional: [20]

$$S[p_\alpha] = -\sum_{\alpha} p_{\alpha} \ln p_{\alpha} - \lambda \left( \sum_{\alpha} p_{\alpha} \mathcal{O}(\alpha) - \langle \mathcal{O} \rangle \right). \quad (1)$$

We find the probability distribution by taking a functional derivative  $\delta S/\delta p_{\alpha}$  and setting it to zero, resulting in:

$$p_{\alpha} = \frac{1}{\mathcal{Z}} e^{-\lambda \mathcal{O}(\alpha)}; \quad \mathcal{Z} = \sum_{\alpha} e^{-\lambda \mathcal{O}(\alpha)}, \quad (2)$$

where  $\lambda$  is the *design pressure*, or the relative importance of the corresponding design objective in driving the distribution. The normalization  $\mathcal{Z}$  is known as the *partition function* and contains a wealth of information on the properties of the *whole* design space. Mathematically, our study of the design problem is reduced to studying how the distribution  $p_{\alpha}$  is affected by soft constraints (design pressure  $\lambda$ ) and hard constraints (the set of available solutions  $\alpha$ ).

*Extracting design information*—with this formulation of design problems as statistical mechanics models, the next challenge is to extract collective properties that encode information about the structure of the design problem and the space it lives in. Doing this typically requires computing sums over a combinatorially large set of  $\alpha$ , which relies on various problem-specific mathematical techniques. For simple scenarios involving only one or two nodes with integrable interactions, prior work has shown that this can be done by coarse-graining to extract effective, so-called Landau, free energies [21]. We expect that effective free energy approaches will, as they do in the ordinary statistical mechanics of particles, provide the means to gain insight into the collective properties of more complex design spaces. However for more complex design spaces, again as is the case in the

ordinary statistical mechanics of particles, some mathematical techniques are required to study systems that lack closed-form, integrable interactions.

To meet the challenge of extracting information about design spaces with complex forms of interaction it is useful to take cues from the structure of the problem. For complex problems the advantage of the logical architecture is that it reduces the complexity of interactions among elements to simple, binary, yes/no connections. The disadvantage of this simplicity is that it loses the richness and specificity of the underlying design problem. This suggests a more complete treatment of the design problem would be to ‘decorate’ the topological description of the logical architecture with information about the ‘topography’ of the underlying design space and its physical architecture. This topographic decoration can be carried out by encoding the design space as a tensor network.

*Tensor network formulation*—tensor networks were originally introduced as a graphic notation for geometric tensors [22], but over the last 25 years have grown into powerful computational tools for storing and manipulating high-rank data. The tensor network computations are especially efficient when the connections are sparse. This property spurred the popularity of tensor networks in a broad range of applications, from encoding entangled wavefunctions in quantum condensed matter systems [23–25], to performing precision quantum chemistry calculations [26], renormalizing lattice models [27, 28], solving constraint counting problems [29, 30], accelerating numerical linear algebra [31–33], and learning multilinear classifiers in machine learning [34]. Across these applications, tensor networks serve as an *information structure* that contains an exhaustive but raw description of the system.

We use tensor networks to bridge the logical and physical descriptions of a design problem space. Network nodes encode the design elements’ topography in the physical space of their placement and their properties. Network connections encode the functional connection topology and topography of the physical interaction of design elements based on their spatial location and physical properties. The topography–topology connection that the tensor network encodes has the useful side benefits that it provides a simple graphical representation of the interaction of design elements, and that well-developed methods exist for extracting information from tensor networks.

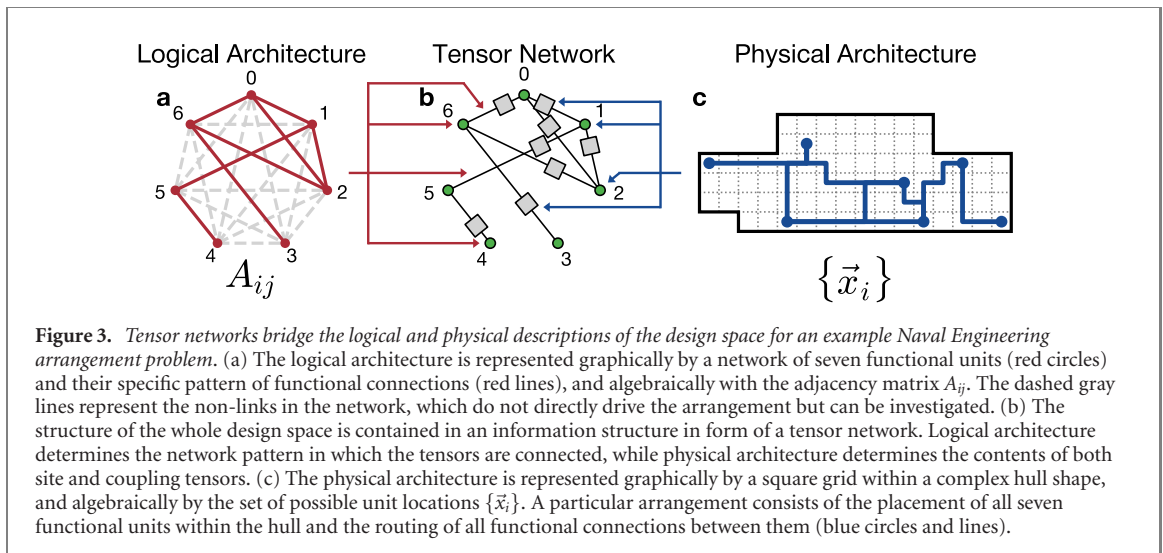
Tensor networks encode information about the design space, but knowledge about patterns of avoidance, adjacency, and association among design elements, has to be extracted with special techniques. Extracting this information requires adding specifically formulated pieces to it that represent key design questions (or, in physics language, observables or order parameters). We formulate design questions about avoidance, adjacency, and association among design elements by acting on the tensor network with a combination of elementary ‘moves’. The moves yield patterns of the placement over sets of solutions in design space, that are computed via contraction of tensor networks. See appendix A for a detailed description of moves and contraction.

We note that recasting the problem in the language of tensor networks is an exact operation. The tensor networks shown on figures below are not qualitative illustrations, but mathematical formulas written in a graphical language explained in detail in appendix A. Though there are many ways to perform and illustrate statistical mechanics computations for systems of simple topology (e.g. well-mixed, lattice, tree), to our knowledge tensor networks are the first method that can deal with arbitrary complex topology. Alternatively, we could evaluate the arrangement patterns by averaging over a representative sample of solutions with Monte Carlo methods, however that would engender statistical and sampling issues that could restrict interpretability. Tensor network computations are free of those issues, at least for the present problem. The numerical contraction computations introduce a controlled approximation error via SVD truncation, but it is of qualitatively different kind than finite sample error.

### 1.3. Example model system

*Functional units and connections*—to demonstrate the systems physics analysis and tensor network computations on a concrete example of a design problem, we first define the specific logical and physical architectures for the problem. We use a problem from Naval Engineering [5], in which the functional units of a shipboard system need to be arranged within the hull of a naval vessel, while respecting their functional connections, such as pipes or cables. The network pattern of connections constitutes the logical architecture, also represented algebraically as an adjacency matrix  $A_{ij}$ . We find a wide variety of network motifs arise in networks of as few as  $n = 7$  functional units without any graph symmetries (figure 3(a)), which we will study in the remainder of this work.

*Ship hull*—we position units and connections within a ship hull that we represent, following [5], by a 2D square grid with a complex, but fixed boundary (figure 3(c)). We constrain all connections within the ship hull to always run along a shortest path between the two functional units; we choose our hull to be  $L_1$ -convex to ensure that at least one shortest path exists between any pair of cells. We find that hull models with a few tens of cells are sufficient to establish placement patterns; computations reported below are for hulls with  $Y_0 = 78$  distinct cells for unit placement each labeled as  $\vec{x}_i$ .



**Figure 3.** Tensor networks bridge the logical and physical descriptions of the design space for an example Naval Engineering arrangement problem. (a) The logical architecture is represented graphically by a network of seven functional units (red circles) and their specific pattern of functional connections (red lines), and algebraically with the adjacency matrix  $A_{ij}$ . The dashed gray lines represent the non-links in the network, which do not directly drive the arrangement but can be investigated. (b) The structure of the whole design space is contained in an information structure in form of a tensor network. Logical architecture determines the network pattern in which the tensors are connected, while physical architecture determines the contents of both site and coupling tensors. (c) The physical architecture is represented graphically by a square grid within a complex hull shape, and algebraically by the set of possible unit locations  $\{\vec{x}_i\}$ . A particular arrangement consists of the placement of all seven functional units within the hull and the routing of all functional connections between them (blue circles and lines).

*Design objectives*—picking the locations of all units and the routings of functional connections between them defines a design solution  $\alpha = (\{\vec{x}_i\}, \text{routing})$ . In early-stage design, design architectures are typically not fixed, therefore the full combinatorial design space needs to be considered. Each design solution is quantitatively evaluated with a *design objective*  $\mathcal{O}(\alpha)$ , here we model routing cost:

$$\lambda \mathcal{O}(\alpha) = \lambda \sum_{ij} A_{ij} C L_1(\vec{x}_i, \vec{x}_j), \quad (3)$$

where  $L_1$  is the ‘Manhattan’ distance between the two cells,  $C$  is the cost per unit distance, and  $\lambda$  is the design pressure. Given the placement of units, we consider all allowed shortest paths between them. By definition, all shortest paths have the same length, so the value of  $\mathcal{O}$  does not depend on the particular routing chosen, yet the number of routings is important. To account for the redundancy of routings, we introduce an *effective design objective*:

$$\lambda \mathcal{O}_{\text{eff}}(\{\vec{x}_i\}) = \sum_{ij} A_{ij} f(\vec{x}_i, \vec{x}_j; T); \quad (4)$$

$$f(\vec{x}_i, \vec{x}_j; T) = \frac{C}{T} L_1(\vec{x}_i, \vec{x}_j) - \ln n_{\text{rout}}(\vec{x}_i, \vec{x}_j), \quad (5)$$

where  $n_{\text{rout}}$  is the number of shortest routings between  $\vec{x}_i$  and  $\vec{x}_j$  within the ship hull, typically growing with distance. The routing lengths  $L_1(\vec{x}_i, \vec{x}_j)$  and the number routings  $n_{\text{rout}}(\vec{x}_i, \vec{x}_j)$  are fully determined by the shape of the hull and can be precomputed, stored as matrices, and scaled by the design pressure as needed.

*Tensor network*—the representation of the design space in form of a tensor network depends on both logical and physical architectures (figure 3(b)). The logical architecture in form of the network  $A_{ij}$  determines the pattern in which the site and coupling tensors are connected. The physical architecture determines the set of available locations for all units  $\{\vec{x}_i\}$  that is used as index for all tensors. The effective design objective  $f(\vec{x}_i, \vec{x}_j; T)$  determines the entries of the coupling tensor. See appendix A for a detailed mathematical discussion.

*Greenfield/brownfield settings*—in the above formulation, the design space  $\{\alpha\}$  of the problem is the space of all possible arrangements of each functional unit  $\{\vec{x}_i\}$ , and is it necessary to establish a means of distinguishing units with fixed and variable position. This distinction is necessary because the formulation needs to address arrangement before or after some of the units have been placed. We refer to situations in which no units have fixed placement as greenfield settings. Greenfield settings are generically associated with green color-coding in results figures that follow. Also, we refer to situations in which one or more units have fixed locations as brownfield settings. Brownfield settings are generically associated with brown color-coding in results figures that follow. In results figures that describe brownfield settings that combine placed units with yet-to-be-placed units we make a visual distinction between the two by color-coding placed units and their effects brown and yet-to-be-placed units green.

*Low cost, high flexibility, and crossover regimes*—the formulation of design problems in terms of spaces of solutions weighted by objectives of the form of equation (5) has been studied in reference [19]. As in reference [19] we expect that the choice of the design pressure associated with each objective (general case: equation (2); this model: equation (3)) will have qualitatively distinct effects on design outcomes [19]. To maximize generality, we study design pressures that correspond to multiple behavioral regimes. We do this by first expressing

the design pressure via its inverse  $\lambda = 1/T$ , where  $T$  is the *cost tolerance*. Low cost tolerance means that minimizing the routing cost  $\mathcal{O}$  is a strong driver of a design solution choice, whereas high cost tolerance means that the choice among the design solution is not driven by cost. Reference [19] showed that the system driven by this design objective undergoes a large-scale rearrangement (akin to a phase transition, but at finite-size) around  $T_{\text{crit}} = C/\ln 2 \approx 1.44C$ . We pick  $C = 1$  to fix the measurement units for  $T$ .  $T < T_{\text{crit}}$  favors low cost and we therefore expect units to organize into motifs that facilitate short (cheap) connecting paths. We expect this setting to be characterized by effective attraction.  $T > T_{\text{crit}}$  favors maximal flexibility and we expect units to organize into motifs that facilitate maximizing routing degeneracy. We expect this setting to be characterized by effective repulsion. We expect that for  $T \approx T_{\text{crit}}$  where cost and flexibility drivers are competing on near-equal footing there will be a crossover in behavior.

## 2. Results

Having formulated our statistical mechanics approach and introduced the model system, we turn to the arrangement patterns of distributed design elements depicted in figure 1 and the quantitative drivers of the patterns of avoidance, adjacency, and association. We find that all three patterns manifest behaviours that are analogous to behaviours observed in condensed matter physics.

The avoidance patterns we operationalize here describe the propensity for functional units in design to preferentially avoid certain regions of space. In our example system, we test this by introducing a void within the ship hull and determining the effect of the void placement on the design objectives.

The adjacency patterns we operationalize here describe the propensity for design elements to arrange into characteristic relative arrangements, or motifs, that persist despite the absolute placement of the elements. In our example system, we compute the distribution of pairwise relative directions between the units and study the effects of topological distance and topology change.

The association patterns we operationalize here describe the propensity for an element introduced into a design to locate preferentially relative to the placement of already-existing functional units. In our example system, we compute the how placement affects design objectives, and find effective forces that drive individual units to preferred locations in a later stage of design.

Our results for each of these three arrangement patterns are described below.

### 2.1. Avoidance

*Void premium*—the interplay of logical and physical constraints among design elements induces a complex landscape for element placement. Intervening in that landscape by reserving space for future use could induce functional units to make a complex, collective rearrangement to avoid the reserved space. We characterize the cost of avoidance by computing the *void premium* that must be paid to forbid any units to be placed in the reserved space.

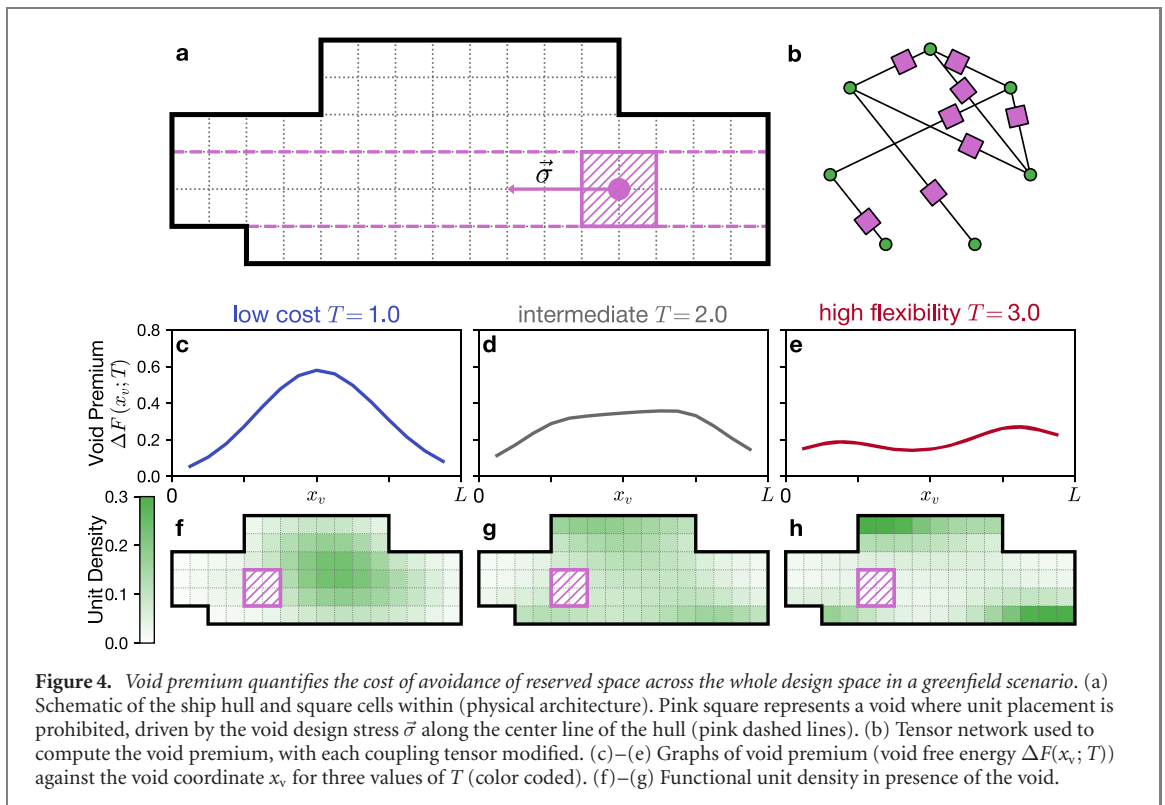
We implement reserved space mathematically by creating a *void*  $2 \times 2$  cells in size. In the results below we introduce the void on the hull midline, though in general it could be placed anywhere. We vary the horizontal location of the void,  $x_v$ , from zero to the hull length  $L$  (see figure 4(a)). We compute the cost of the void via the tensor network approach by suppressing several rows and columns of the coupling tensor (see appendix A). Contracting the modified tensor network results in a modified partition function, which is smaller or equal to the original one  $\mathcal{Z}(x_v; T) \leq \mathcal{Z}(T)$ . The ratio of the two partition functions defines the non-negative free energy:

$$\Delta F(x_v; T) = -\ln \frac{\mathcal{Z}(x_v; T)}{\mathcal{Z}(T)}. \quad (6)$$

We take this void free energy as a measure of the void premium, the effective ‘cost’ of the avoidance of a specified region in space.

*Void design stress*—the magnitude of the void premium  $\Delta F$  corresponds to the placement opportunity cost for the functional units. To understand this opportunity cost, note that functional units that are not yet placed form a greenfield ‘cloud’ of possibilities within the hull, the location and density of which depends on the cost tolerance  $T$ . Cutting a void from a dense part of the cloud costs a lot of free energy, whereas cutting a cloud from a sparse part of the cloud costs almost nothing. In this way, scanning the void free energy along the void coordinate  $x_v$  gives a direct probe of the morphology of the cloud. Conversely, if we regard the unit positions as fixed, and the void as moveable, the cloud of units drives the void with an effective force  $\sigma = -\Delta F/\Delta x_v$ , which we call *void design stress*. The void free energy and design stress are then a concise description of the collective effect of avoidance in functional unit placement.

The void premium and void design stress give a description of collective avoidance effects in placement. These effects can be examined in greenfield and brownfield settings.



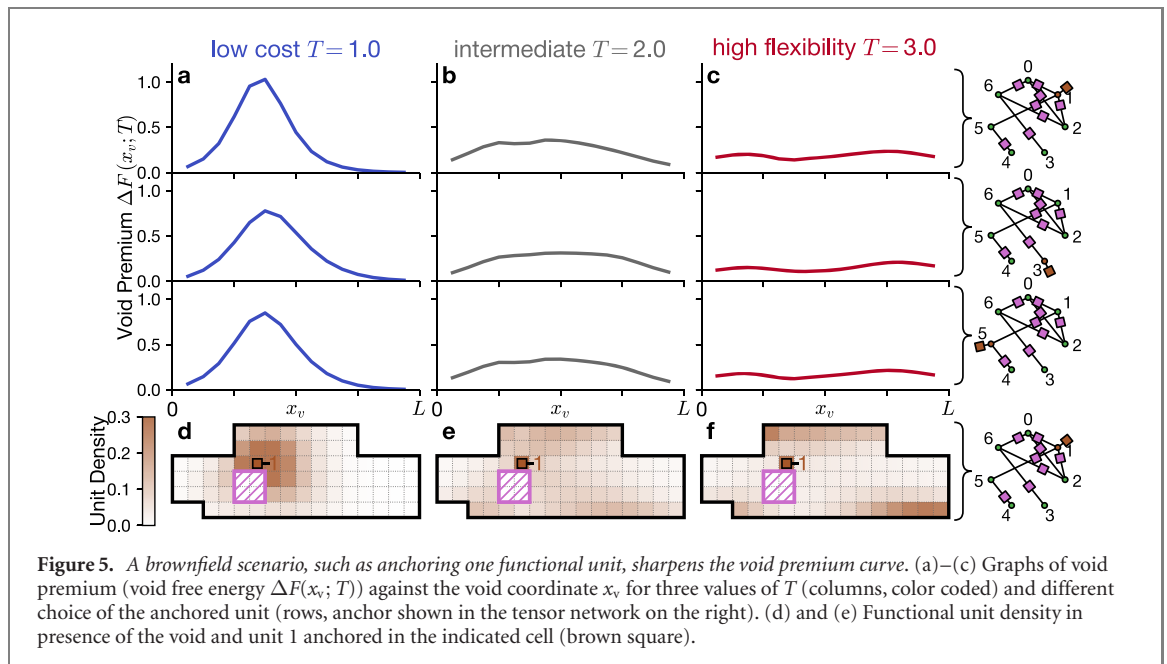
*Greenfield*—we studied avoidance metrics in greenfield settings, i.e. before any unit data have been fixed, in three design regimes specified by cost tolerance  $T$ . We plot results in figures 4(c)–(e). At subcritical  $T = 1.0$  (low cost priority), the void free energy curve shows a clear single maximum in the middle of the hull, and two minima on the ends of the hull. At near-critical  $T = 2.0$  (cost-flexibility tradeoff) the curve maintains the same qualitative shape, but the maximum gets flatter. At supercritical  $T = 3.0$  (high flexibility priority), the curve shape flips to have a local minimum at the center of the hull, surrounded by local maxima on two sides. In other words, at low  $T$  the void prefers to be at either of the two ends of the ship (but a choice needs to be made in favor of one of them). In contrast, at high  $T$  the void prefers to be in the center of the ship. Thus the change from designing for flexibility (high  $T$ ) to designing for cost (low  $T$ ) induces a change from one architecture class (central-void) to two architecture classes (bow-void and stern-void). This collective effect is analogous to symmetry-breaking phase transitions in conventional physical systems [35].

To understand the origin of the symmetry breaking we note that void free energy is a proxy for the morphology of the unit cloud. To illustrate the shape of the cloud in a different way, we approximate the cloud density as a sum of one-unit densities  $\rho(\vec{x}) = \sum_i p_i(\vec{x})$  and plot densities as heatmaps in figures 5(f)–(h). These heatmaps are approximate because, unlike the void free energy curves, they ignore the correlations in unit placement. At low  $T$  (panel (f)), units attract each other and thus preferentially form a cloud in the center of the hull and push the void to either side of the hull. At near-critical  $T$  (panel (g)), the distribution becomes more homogeneous throughout the hull, flattening the curve. At high  $T$  (panel (h)), functional units strongly repel one another, concentrating near the edges of the hull. This leaves the center nearly empty, resulting in a single void free energy minimum.

*Brownfield*—both the void free energy curve and the unit cloud morphology can, however, change dramatically in brownfield settings, e.g. if even one unit is fixed to a specific location in space. We pick the location indicated with the brown square in figures 5(d)–(f) and fix one unit there. We choose three different units to fix: unit 3 (which has 1 functional connection), unit 5 (2 functional connections) and unit 1 (3 functional connections). We plot the resulting void free energy curves in panels (a)–(c), and unit clouds in panels (d)–(f).

Consider first the low-cost regime  $T = 1.0$  (figures 5(a) and (d)). In the greenfield setting (figure 4(f)) units positions were determined solely by ship geometry, and formed a dense cloud in the middle of the ship (figure 4(f)). Fixing a unit position places an additional constraint on unit positions, and forces the unit cloud to condense around it (figure 5(d)). Because of this condensation, the void free energy curve becomes simultaneously steeper and more focused around the fixed unit point (panel (a)), but decays faster close to the edges of the hull. The void free energy cost also depends on the topological position of the anchored unit: it is highest for the most-connected unit 1 (bottom curve) and lowest for the least-connected unit 3 (top curve).





At the near-critical and supercritical  $T = 2.0, 3.0$  the anchored unit similarly creates a reference point for the cloud, but the units in the cloud *repel* from that point. When repulsion and attraction are nearly balanced at  $T = 2.0$ , the cloud profile becomes nearly uniform and is not strongly affected by the fixed unit (compare figures 4(g) and 5(e)). Similarly, fixing a unit at supercritical  $T = 3.0$  creates a point of strong repulsion, forcing the unit cloud to the opposite corners of the ship hull (figures 4(h) and 5(f)). At both values of  $T = 2.0, 3.0$ , the cloud morphology is not affected strongly by the single fixed unit position, and thus the brownfield void premiums (figures 5(b) and (c)) closely resemble their greenfield counterparts (figures 4(b) and (c)). Discussion of the effects of avoidance on unit positions, i.e. backreactions on the cloud, can be found in appendix B.

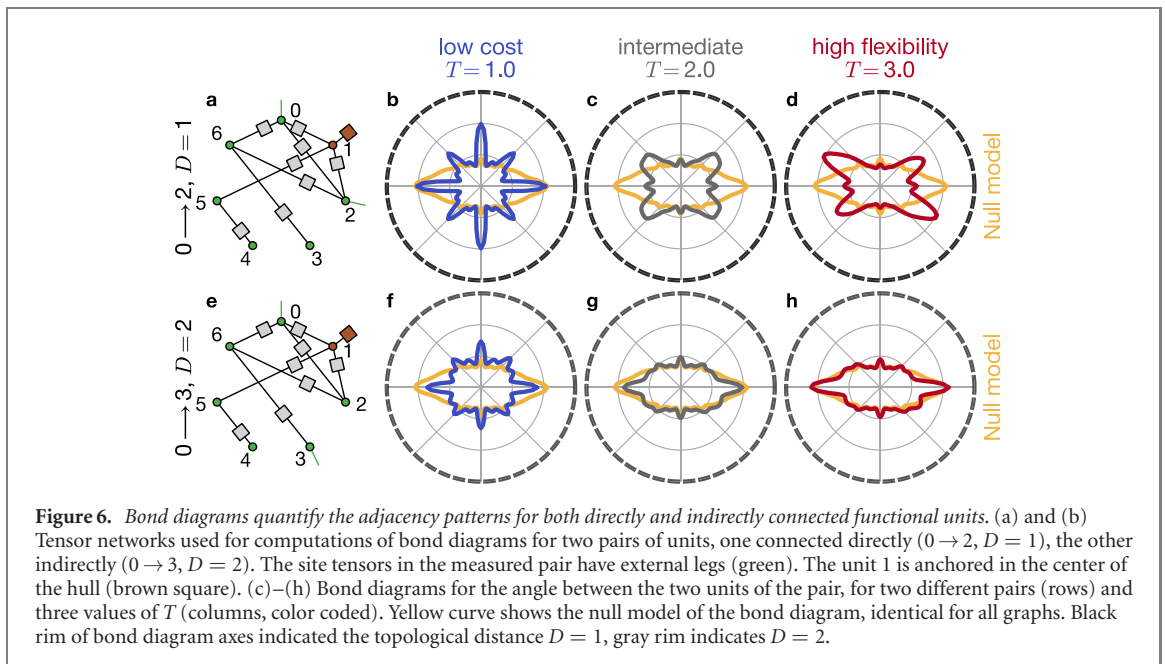
*Avoidance: logical–physical architecture interplay*—the above avoidance analysis gives a case study of basic phenomenology of the interplay between design pressure (favoring low-cost vs high-flexibility) and the logical and physical architecture. Shifting the design priority from low-cost to high-flexibility changed the interaction between pairs of functional units. However, unit interactions were modulated by connection topology (i.e., logical architecture) and by the spatial domain (i.e., the physical architecture). We captured the effect of these complex interactions on spatial avoidance the unit clouds in figures 4 and 5. However, within the unit clouds, the interplay of design pressure with the logical and physical architectures also induces emergent coupling. This emergent coupling within the cloud induces patterns of adjacency and association between units, which we turn to next.

## 2.2. Adjacency

Whereas our avoidance analysis derives from and illustrates the basic morphology of the unit cloud within the ship hull, questions about unit adjacency derive from correlations within the cloud, and the emergent coupling between units that arise.

*Bond-diagram measures of adjacency*—to determine how emergent coupling between units leads to arrangement motifs, we consider pairs of units and their relative positions. We examine pairs of units in 2D space, and express motifs as the polar angles  $\theta(i \rightarrow j)$ , which vary along with the positions of the units in the cloud. Across the cloud, the angle takes form of a probability distribution  $p_{i \rightarrow j}(\theta)$ . This distribution is analogous to the bond order that is used to describe structure in condensed matter [36, 37]. In condensed matter, bond angles are whole-system aggregate measures of adjacency. Here, the heterogeneous connectivity of design elements yields ‘bond’ diagrams that are specific to each pair of units  $i, j$ . Depending on whether the units  $i$  and  $j$  are directly connected or not ( $A_{ij} = 1$  or 0), the bond diagrams illuminate the strength of direct or emergent adjacency patterns.

*Computational approach*—to compute the bond diagrams mathematically, we use tensor networks to compute the raw two-unit marginal distributions  $p(\vec{x}_i, \vec{x}_j)$  (figures 6(a) and (e)), and convert them into the angular distributions  $p_{i \rightarrow j}(\theta)$  using Kernel density estimation (KDE) to reduce the numerical artifacts (see appendix A). In order to demonstrate more sharply defined bond diagrams, we assume that one unit has already been placed (anchored) in the center of the hull and all other units need to be placed with respect to it. The bond diagram  $p_{i \rightarrow j}(\theta)$  of any pair of units is not uniform with respect to the angle  $\theta$  even if the units are not connected at all,

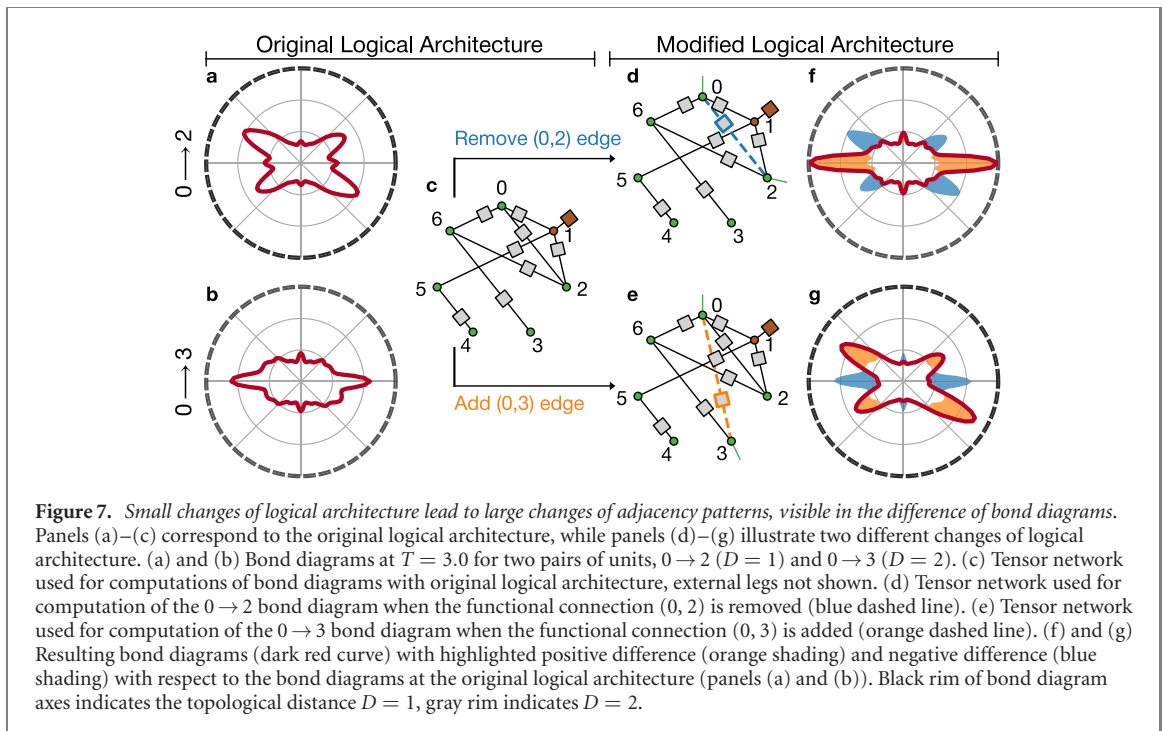


directly or indirectly. This non-uniformity is driven by the shape of the hull, a manifestation of the physical architecture, and we account for it by computing *null model*  $p_0(\theta)$  of the bond diagram (see appendix A). Differences between the null model and computed bond diagrams are indicators of interaction-driven adjacency. This interaction driven adjacency depends on unit connectivity; we use a *topological distance*,  $D(i, j)$  metric.  $D(i, j)$  is the minimal number of network hops to get from unit  $i$  to unit  $j$ . In our example problem, the minimal number of hops varies from 1 (e.g. units  $0 \rightarrow 1$ ) to 5 (e.g. units  $3 \rightarrow 4$ ).

*Direct adjacency*—we show topological distance, bond diagrams, and the null model for our model in figure 6 in form of polar plots for two example unit pairs (corresponding plots for all unit pairs are given in appendix B). We start discussion with the bond diagrams for direct adjacency ( $0 \rightarrow 2$ ,  $D = 1$ ). At subcritical  $T = 1.0$  (panel (b)) most units are located very close to each other, either in cardinal or intercardinal directions (orthogonally or diagonally), resulting in a bond diagram with a strong eightfold signal. At near-critical  $T = 2.0$  (panel (c)) the orthogonal attraction is balanced with diagonal repulsion, resulting in a bond diagram with smaller peaks. At supercritical  $T = 3.0$  (panel (d)) the units are located relatively far from each other and prefer diagonal relative location (since diagonal location allows them to maximize their routing entropy), resulting in a fourfold, X-shaped signal. The symmetry of the fourfold signal is further broken by anchoring the unit 1. This additional symmetry breaking is driven by the high density of units in top-left and bottom-right corners of the hull (see figure 4).

*Emergent adjacency*—across the whole  $T$  range, the bond diagrams for direct adjacency are significantly different from the null model. However, adjacency can also be induced for indirectly connected unit pairs. The indirectly connected unit pair  $0 \rightarrow 3$  shows emergent adjacency, since its bond diagram is different from both the direct adjacency and the null model (figures 6(f)–(h)). The bond diagrams for unit pairs with even larger topological distance the bond diagrams gradually approach the null model (see appendix B), following the intuition of decay of correlation functions with distance in condensed matter systems. This observation suggests the general takeaway: at large topological distance bond diagrams always approach the null model, and thus are fully determined by the physical architecture; at small topological distance the bond diagrams depend strongly on both the design pressure  $T$  and the explicit details of logical architecture. However, our results reveal that for emergent adjacency topological distance is a good predictor of *strength* but is not a good predictor of *shape*.

*Logical architecture modifications*—to further test the interplay between logical architecture and adjacency, we investigate what happens to bond diagrams when we modify the logical architecture. We consider two types of modifications: removing an existing functional connection (figure 7(d)), or adding a new one between two units (figure 7(e)). Instead of comparing the resulting bond diagrams to the null model again, we focus on the difference between bond diagrams before and after modification (panels (f) and (g)). *Addition*: the effect of adding the (0, 3) connection is strong. We can anticipate that with the added direct adjacency, the adjacency pattern should approach that of other directly adjacent units. The fourfold signal that results (panel (g)) is a signal of this. Since neither of the units 0 or 3 is explicitly anchored in space, at high  $T = 3.0$  they want to be positioned at the opposite ends of the longest diagonal available within the hull, in this case the diagonal



from top-left to bottom-right corners, similarly to the original adjacency  $0 \rightarrow 3$  (figure 6(d)). *Removal:* like for addition, the effect of removing the  $(0, 2)$  connection is dramatic, however the result is unexpected. Instead of fourfold diagonal direct adjacency, the two units now have twofold horizontal emergent adjacency (panel (f)). The reason for this is that with the  $(0, 2)$  connection removed, the units  $1-2-6-0$  now form a rhombus. Unit 1 is fixed in space, and because of high  $T = 3.0$  all unit pairs prefer to have diagonal adjacency. In this case the units 0 and 2 on the opposite corners of the rhombus will have an orthogonal adjacency, of which only horizontal adjacency manifests because the ship hull is larger in length than height.

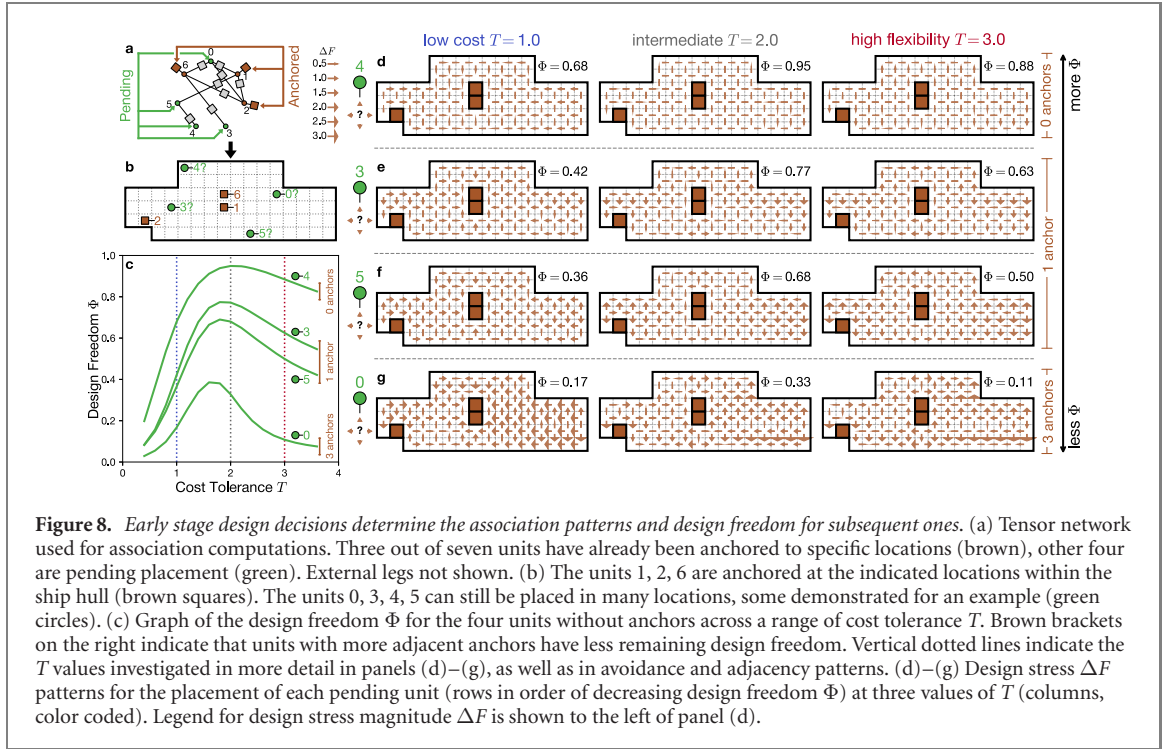
*Adjacency: changes and constraints drive patterns*—we showed that the irregular, complex-network nature of a system’s logical architecture drives the patterns of direct and emergent adjacency. We showed adjacency patterns can change significantly with changes in logical architecture. One outcome of this approach was the ability to detect emergent adjacency. The emergent effects we observed were with a single fixed unit. Though having few fixed units is a characteristic of early-stage design, later-stage design situations will result in more fixed units. Fixing more units will induce more constraints, and further constraints will complicate the interplay of the logical and physical architecture. A more complicated logical–physical architecture interplay should induce more complex patterns of association between units, which we will examine next.

### 2.3. Association

Analyzing association patterns extends our avoidance and adjacency investigations to situations in which multiple, pre-existing constraints restrict functional units, i.e. in brownfield settings. These settings model either of two situations: (i) actual late-stage design in which multiple functional units have been fixed during preceding design stages, or (ii) an early-stage design investigation of hypothetical late-stage situations under different decision scenarios.

*Constraints and localization*—in either case, the expectation that multiple active constraints will drive complex forms of interaction suggests that identifying patterns of association that result will require different techniques than identifying patterns of avoidance and adjacency. In general we expect that patterns of association arising from multiple constraints will localize those patterns relative to fixed design elements. This suggests that metrics of association patterns should signal a tendency toward (or away from) placement proximity relative to fixed elements, either globally or locally. Here, for a global signal we adapt measures of emergent localization to compute a scalar design freedom for unit locations. For a local signal we compute the design stress associated with specified, hypothetical unit placement.

To study the effect of added constraints, their interplay with the logical and physical architecture and the resulting localization, we employ the same model system as in the avoidance and adjacency investigations. However we introduce constraints that fix units 1, 2, and 6 to specific locations. We investigate the emergent localization of the other 4 units with two metrics via the global design freedom  $\Phi$  and local design stress  $\Delta F$ . Results are shown in figure 8 and broken down below.



*Global signal of association: design freedom*—mathematically, both global and local metrics of localization use a tensor network computation of the marginal probability distribution  $p(\vec{x}_i)$ . The conversion of the distribution into design freedom is inspired by the metric of existence area, commonly used in studies of the Anderson localization of wavefunctions in disordered media and the localization of vibrational eigenmodes [38, 39]. We define design freedom as:

$$\Phi = \frac{1}{Y_0} \frac{\left( \sum_{\vec{x}} p(\vec{x}) \right)^2}{\sum_{\vec{x}} p(\vec{x})^2}, \quad (7)$$

where the normalization  $Y_0$  is the total number of cells within physical architecture; in this example  $Y_0 = 78$ . Given this normalization,  $\Phi$  takes a value between 0 and 1 and has the meaning the effective fraction of the total area available for unit placement, if the distribution was uniform. For a unit with uniform distribution  $p(\vec{x}_i) = \text{const.}$ ,  $\Phi$  would be 1, whereas for an anchored unit  $\Phi$  would be  $1/Y_0 \rightarrow 0$ .

Because of the heterogeneous connectivity of the logical architecture, design freedom  $\Phi$  varies between units. The variation between units is in addition to variation with design pressure, via changing cost tolerance  $T$ . Figure 8(c) plots  $\Phi(T)$  by unit, and shows that all units have design freedom peaks near  $T \approx 2.0$ . In the range of this near-critical  $T$ , cost (effective attraction) and flexibility (effective repulsion) drivers of unit interactions balance and allow the units to explore the largest range of placement. As well, we observe  $\Phi(T)$  to fall into three groups according to how constrained each unit is. Unit 4 is not directly connected to any of the anchored units and thus enjoys the largest design freedom, almost approaching  $\Phi = 1$ . Units 3 and 5 are each connected to one anchored unit and thus have intermediate  $\Phi$ . Unit 0 is connected to all three anchored units and thus has the lowest  $\Phi$  which quickly decays at both low and high  $T$ .

*Local signal of association: design stress*—whereas  $\Phi$  serves as an global scalar metric of design freedom, it is also important to understand how global design freedom is distributed locally. This local distribution is captured by design stress. Design stress is closely related to an effective (Landau) free energy (LFE), defined as follows:

$$F(\vec{x}_i) = -\ln p(\vec{x}_i) + C, \quad (8)$$

where  $C$  is an arbitrary additive constant. We chose a convention where  $C$  is such that the minimal value of  $F$  is zero. The LFE can be interpreted as an effective design objective for the chosen degree of freedom, given that other degrees of freedom have been fixed or integrated out. This interpretation is analogous to the void premium (equation (6)) in our avoidance investigation, but instead of the effects of unit placement on voids, here we examine the effects of unit placements on one another. Similar to the definition of void design stress via a spatial difference, the difference of LFE between two horizontally or vertically adjacent cells is the design stress  $\Delta F$ . Design stress is then an effective ‘force’ that pushes individual functional units towards their

preferred locations [19, 21]. Compared with global design freedom, design stress patterns give a more detailed picture of effective localization.

Figures 8(d)–(g) presents the design stress patterns for all four pending units at three values of  $T$ . Design stress is represented by brown arrows drawn across the boundary of two adjacent cells and pointing from higher to lower LFE. In other words, to decrease LFE and reach lower values of its effective design objective, a unit needs to follow the arrows towards a basin. As the basin gets smaller and its walls get steeper, the pending units become more closely associated with the anchored ones and thus exhibit stronger emergent localization. The localization effect is strongest at lowest  $T$ , where all of pending units are strongly attracted to the two anchored units 1, 6 in a single basin. The basin is steepest for the most constrained unit 0, less steep for the units 3, 5, and the shallowest for the least constrained unit 4, consistent with our expectation based on design freedom  $\Phi$ . At higher  $T = 2.0, 3.0$ , the constrained unit 0 develops complex LFE and design stress landscapes with multiple local minima, maxima, and ridges (panel (g)). Units 3, 5, connected respectively to the anchored units 6 and 1 in the center of the hull, show an X-shaped pattern of LFE (panels (e) and (f)), similar to the bond diagrams for directly connected unit pairs, e.g. figure 6(d). Lastly, the unit 4 is not connected to any of the anchored units, instead it is ‘dangling off’ unit 5 and thus shows almost nonexistent design stress across the whole hull (panel (d)).

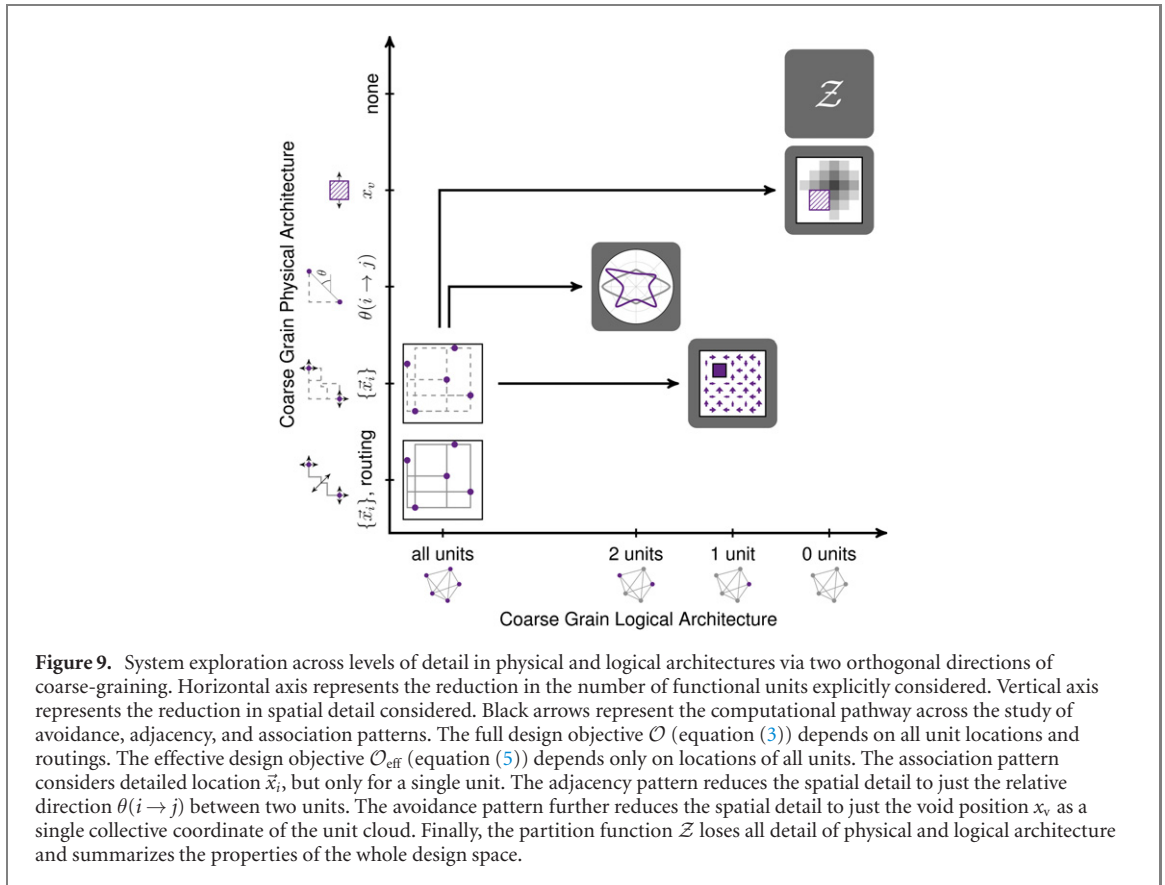
*Association: interaction and decision drivers*—both the design stress and design freedom metrics show that the association patterns and the emergent localization phenomenon strongly depend on the position of both the fixed and the pending units within the logical architecture. The logical architecture alone gives an interpretation of the emergent localization result by counting the anchored neighbors. However, fully predicting localization requires examining the logical–physical architecture interplay that arises from the systems physics analysis. Unlike the simplified unidirectional design stress discussed in reference [21], in this system the design stress pattern is emergent both from unit interactions and from previous design decisions. Chaining design decisions into sequences and achieving optimal control of emergent localization stands out as an important question for further study.

### 3. Discussion

In this paper we showed that questions of avoidance, adjacency, and association among the elements of complex, distributed systems hinge on the interaction between logical- and physical-architecture description planes. We bridged these descriptive planes with statistical physics techniques and showed that patterns of avoidance, adjacency, and association can be mapped for an example system.

*Design phenomena: symmetry-breaking, emergent adjacency, localization*—our mapping of avoidance gave a space premium landscape. We found this landscape to undergoes a symmetry-breaking transition with a change from design pressure that prioritizes high flexibility to pressure that prioritizes low cost (figures 4 and 5). Our mapping of adjacency gave a description analogous to ‘bond’ directions in matter systems. From this bonding description we observed that indirectly connected design elements developed emergent adjacency (figure 6). We also found large downstream changes in adjacency from small changes in underlying connectivity (figure 7). Our mapping of association patterns quantified changes in global design freedom driven by fixing design elements and changes in design pressure. Mapping these effects locally showed the emergent localization of design elements (figure 8).

*Coarse graining for other design contexts*—our mappings of avoidance, adjacency, and association patterns were done for a model system motivated by problems in Naval Engineering. However, for other design contexts where questions of avoidance, adjacency, and association patterns arise, our statistical physics approach opens new lines of attack. In particular, our approach can be summarized in two steps. First we ‘decorated’ the logical architecture with detail from the physical architecture. Then, we systematically chose two sets of system details, one to examine in detail, and the other to treat in aggregate, in a coarse-grained way. The aggregated details induce effective patterns of interaction among the remaining elements, that reveal underlying patterns of arrangement. We illustrate this strategy in figure 9. Figure 9 casts the strategy into two orthogonal forms of coarse-graining: one in the physical architecture, the other in the logical architecture. In this representation, in statistical physics language, microstates that retain complete detail of both the physical and logical architecture sit in one corner, whereas the partition function, which aggregates microstates into a single scalar sits in the opposite corner. Though the specific locations that correspond to our investigations are given at specific points on these axes, regardless of design context, answers to questions about patterns of avoidance, adjacency, and association lie at intermediate levels of detail between those extremes.



## Acknowledgements

We thank C X Du for useful discussions, A S Jermyn for extensive help with the PyTNR package, and N Mackay for assistance in development of the TenZ package. This work was supported by the US Office of Naval Research Grant Nos. N00014-17-1-2491 and N00014-15-1-2752. GvA acknowledges the support of the Natural Sciences and Engineering Research Council of Canada (NSERC).

## Data availability statement

The data that support the findings of this study are available upon reasonable request from the authors.

## Appendix A. Methods

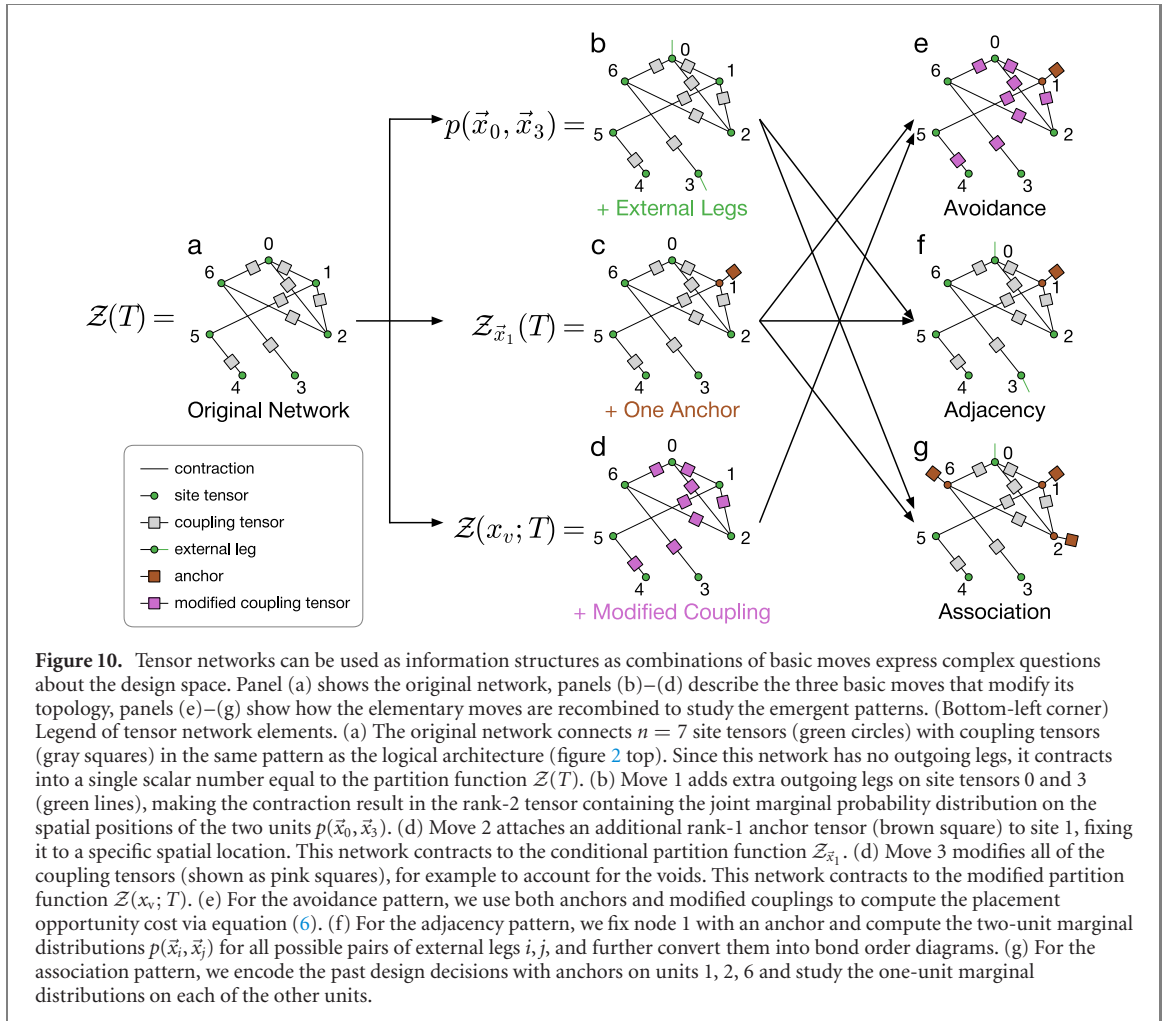
### A.1. Tensor network construction

The partition function of the system can be expressed in terms of the effective design objective in the following factorized form:

$$\mathcal{Z}(T) = \sum_{\{\vec{x}\}} \prod_{\substack{i < j \\ A_{ij} \neq 0}} e^{-f(\vec{x}_i, \vec{x}_j; T)}. \quad (\text{A1})$$

We interpret the factorized partition function (A1) graphically in the form of a tensor network (figure 10(a)). Like other networks, a tensor network consists of nodes and links. Each node is a tensor with rank equal to node degree, and the network links represent the pattern of tensor contractions. We construct tensor networks following a recent prescription of reference [40], in which the tensor networks are bipartite: they consist of *coupling tensors* and *site tensors*, with each type only connected to the other type as shown in figure 10(a). Logical architecture  $A_{ij}$  is contained in the pattern of the coupling tensors, whereas physical architecture domain  $\{\vec{x}\}$  serves as the index of all the tensors, and the design objective determines the contents of the tensors.

The basic tensor network (figure 10(a)) represents the partition function  $\mathcal{Z}(T)$  and is merely an alternative, graphical way to express equation (A1). Each multiplicative term in the partition function becomes a rank-2 coupling tensor with elements defined as  $M_{\vec{x}_i, \vec{x}_j} \equiv e^{f(\vec{x}_i, \vec{x}_j; T)}$  (note elementwise rather than matrix exponentiation). The site tensors have the form of a multi-dimensional Kronecker delta with the rank corresponding to



site's degree in the logical network  $A_{ij}$ . For example, the unit 0 in figure 10(a) has three network neighbors, and thus corresponds to a rank-3 tensor  $\delta_{\vec{x}_{01}\vec{x}_{02}\vec{x}_{06}}$ . The indices mean the value of unit 0 coordinate  $\vec{x}_0$  that is 'presented' to each network neighbor, in this case units 1, 2, and 6. The Kronecker delta ensures that each neighbor perceives the unit 0 at the same location, while index summation ensures that all possible locations are considered. The rank of site tensors can be adjusted for other computations, as shown below. For brevity of notation, we suppress indices on edges because the contraction pattern is dictated by the graphical notation.

Contracting all the tensors along the network links corresponds to performing the sum in equation (A1). Since that sum has no free indices, the network of figure 10(a) has no external (unpaired) legs. Contraction of the network preserves the number of external legs, resulting in a rank-0 tensor, a scalar number. Since each coupling tensor implicitly depends on cost tolerance  $T$ , the result of contraction is the  $T$ -dependent partition function  $\mathcal{Z}(T)$ .

In order to compute quantities other than the partition function, we add minor modifications of the tensor network. These modifications are described in the graphical language of moves. Here we define three moves: adding external legs, adding anchors, and modifying the coupling tensors (figures 10(b)–(d)). These moves are recombined to create networks that address the patterns of avoidance, adjacency, and association (figures 10(e)–(g)).

## A.2. Move 1: external legs

The first move adds extra legs to specific site tensors to control whether specific design degrees of freedom are marginalized or not. If none of the degrees of freedom are marginalized, then carrying out the multiplication but not the summation in the sum (A1) would result in an un-normalized joint probability distribution  $p(\{\vec{x}_i\})$  over all the units, which is a rank- $N$  tensor of prohibitive size. However, following the usual probability theory calculus, in a joint probability distribution each of the entering variables can be in three states: joint, marginalized, or conditional. In the tensor network representation of figure 10(a), every variable  $\vec{x}_0, \vec{x}_1, \dots$  is marginalized, resulting in the distribution normalization, i.e. the partition function  $\mathcal{Z}(T)$ .

In this perspective, a special action needs to be taken to *not* marginalize some of the variables. We do this by adding external legs to the corresponding site tensors (green lines in figure 10(b)). External legs are the

site degrees of freedom that are *not* summed over, functioning as free indices for the sum (A1). The result of contracting the network in figure 10(b) is a rank-2 tensor that represents the un-normalized joint probability distribution  $p(\vec{x}_0, \vec{x}_3)$ . Since the original network contracted to yield the full  $\mathcal{Z}$ , the normalized probability distribution can be expressed as  $\tilde{p} = p/\mathcal{Z}$ .

### A.3. Move 2: anchors

The second move adds anchor tensors to the network. The anchors represent the design decisions already taken and woven into the information structure, thus encoding the brownfield aspects of design. In tensor network language, this is equivalent to fixing some of the local degrees of freedom  $\vec{x}_i$  and thus summing over a restricted ensemble, *conditional* on the fixed  $\vec{x}_i$ . We do this by creating an additional tensor that we call an ‘anchor’. An anchor is a rank-1 tensor (vector) that is coupled to a site tensor and is illustrated as purple square in figure 10(c). The elements of an anchor vector are given by the Kronecker delta  $\delta(\vec{x}_i, \vec{x}_a)$ , where  $\vec{x}_i$  is the index connected to the site and  $\vec{x}_a$  is the specific location to which the functional unit is pinned as result of a design decision. Since we did not create any external legs, the tensor network in figure 10(c) also contracts to a scalar number of *conditional* partition function  $\mathcal{Z}_{x_a}$  that functions as a similar statistical summary of the system as the original partition function  $\mathcal{Z}$ .

### A.4. Move 3: modified coupling

The third move modifies the coupling tensors  $M(\vec{x}_i, \vec{x}_j)$  and traces the effect of this modification on the partition function. In our study, we use the modification to account for the void where no units can be placed. The modification consists of suppressing the statistical weight of the void cells in the coupling tensor:

$$M^*(\vec{x}_i, \vec{x}_j) = M(\vec{x}_i, \vec{x}_j) \prod_{\vec{x}_v} (1 - \delta(\vec{x}_i, \vec{x}_v))(1 - \delta(\vec{x}_j, \vec{x}_v)), \quad (\text{A2})$$

where  $\vec{x}_v$  denotes the positions falling into the excluded void. In the network on figure 10(d) we modified each coupling tensor in this way (marked in pink). The network results in the modified partition function  $Z(x_v; T)$ , from which we compute the void free energy via equation (6).

### A.5. Bond diagrams

To compute bond diagrams, the raw two-unit distributions  $p(\vec{x}_i, \vec{x}_j)$  need to be converted into the angular distributions  $p_{i \rightarrow j}(\theta)$  in post-processing. Since all functional units are placed in a discrete, finite, and fixed set of cells  $\vec{x}_i$ , we pre-compute the directions between any pair of cells  $\theta(\vec{x}_i, \vec{x}_j)$ , measured in radians from 0 to  $2\pi$ , ahead of time and store them.

Within the design ensemble, the locations of units  $\vec{x}_i$  are random, drawn from the joint distribution encoded in the tensor network. We compute a series of marginal two-units distributions  $p(\vec{x}_i, \vec{x}_j)$  for all pairs  $i < j$  (see the tensor network in figure 10(f)). Since the possible unit locations are discrete, the possible directions  $\theta(\vec{x}_i, \vec{x}_j)$  form an artificially irregular discrete set. This numerical artifact would result in a jagged direction distribution  $p_{i \rightarrow j}(\theta)$ . We smooth the distribution by using a version of non-parametric KDE [41, 42] with periodic boundary conditions in which higher angular harmonics are suppressed:

$$p_{i \rightarrow j}(\theta) = \frac{1}{\mathcal{N}} \sum_k \sum_{\vec{x}_i, \vec{x}_j} p(\vec{x}_i, \vec{x}_j) e^{-\frac{1}{2}(hk)^2} \cos(k(\theta - \theta(\vec{x}_i, \vec{x}_j))) \quad (\text{A3})$$

Here  $\mathcal{N}$  is a normalization factor,  $h$  is the KDE smoothing factor (bandwidth),  $k \in \{0, 1, 2, \dots\}$  is the angular mode index. We find that using smoothing factor of  $h = 0.1$  radians and angular modes up to  $k_{\max} = 30$  gives good results.

The resulting distributions  $p_{i \rightarrow j}(\theta)$  need to be compared with the null distribution induced by the physical architecture (ship hull shape). We compute the null distribution by evaluating the formula (A3) for  $p(\vec{x}_i, \vec{x}_j) = \text{const.}$ . The identical null distribution is shown in every panel of figure 6.

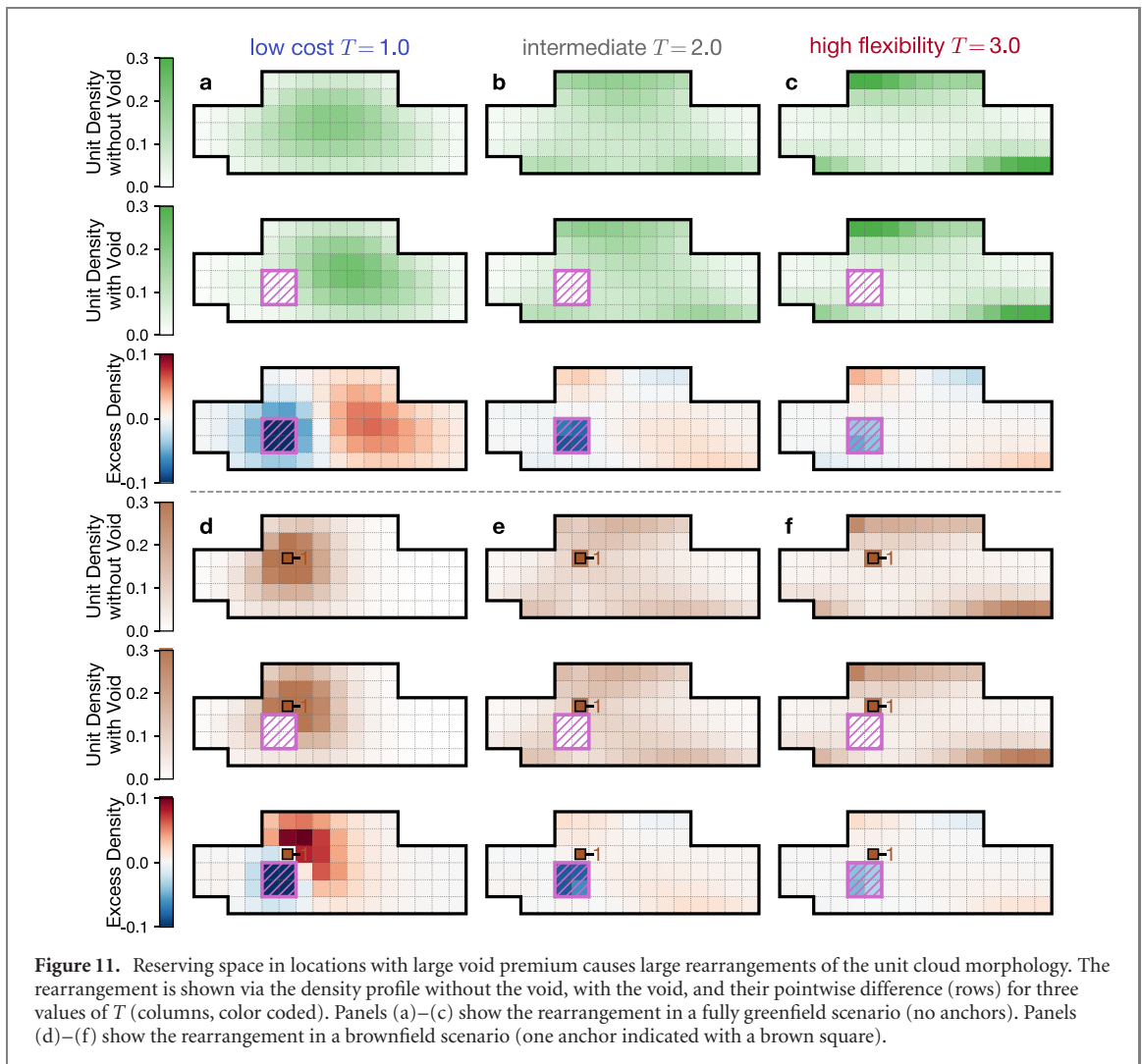
### A.6. Numerical aspects of computation

Implementing the tensor networks described above on a computer requires two different kinds of computational work: constructing the networks from logical and physical architecture and possible modifications with the three moves, and contracting said networks numerically. We perform these two tasks in Python.

Using code we developed, we create tensor networks by specifying the network topology, the spatial domain geometry, the design objective, and additional moves. These specifications are done via high-level commands, allowing for the rapid generation of diverse networks.

Tensor network contraction is handled by a Python package. Existing tensor network packages use different methods of executing a sequence of pairwise tensor contractions. The contraction result does not depend



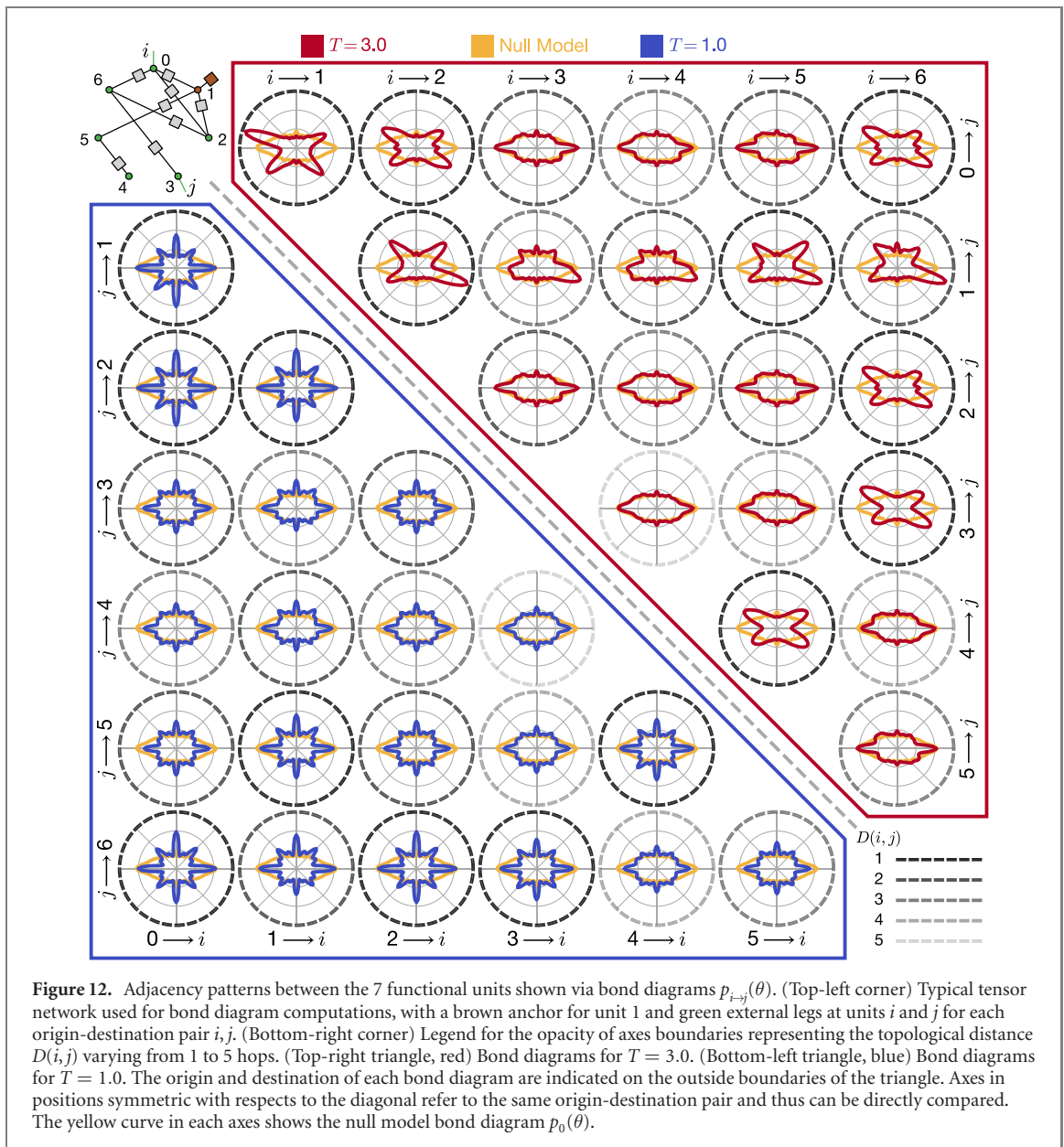


**Figure 11.** Reserving space in locations with large void premium causes large rearrangements of the unit cloud morphology. The rearrangement is shown via the density profile without the void, with the void, and their pointwise difference (rows) for three values of  $T$  (columns, color coded). Panels (a)–(c) show the rearrangement in a fully greenfield scenario (no anchors). Panels (d)–(f) show the rearrangement in a brownfield scenario (one anchor indicated with a brown square).

on the contraction sequence, but the computational time and memory requirements rise by orders of magnitude for suboptimal sequences. Optimal sequences are known for certain frequently used networks, whereas for others one can use exhaustive enumeration algorithms to find the optimal sequence and then execute it repeatedly for the same network topology [43]. However, almost all networks that we contract in the present work are subtly different, and therefore might require different contraction sequences. We perform all contractions with the PyTNR package, an open-source general purpose tensor network contractor [40, 44]. The features of PyTNR include using heuristics to automatically generate the contraction sequences on the fly, and performing SVD approximations of controlled precision to reduce the dimension of stored tensors.

The features of PyTNR define the computational constraints on the size of systems that our approach can handle. The size and structure of the logical architecture directly change the number of units  $n$  in the network and the number of tensors  $n_t$  (counting both site and coupling tensors). The size or resolution of the physical architecture domain directly affect the tensor bond dimension  $D$ . A rigorous, though pessimistic upper bound on the time complexity of contraction stands at  $\mathcal{O}(D^{2\sqrt{\Delta}n_t})$ , where  $\Delta$  is the maximal tensor rank [30]. In contrast, PyTNR relies on a heuristic and stochastic generation of contraction sequences, which complicates even the empirical investigations of numerical scaling of complexity. For highly structured networks, such as  $d$ -dimensional hypercubic lattices, the time complexity scales a power law  $\mathcal{O}(N^\gamma)$ , where the exponent  $\gamma$  is a bit larger than the space dimension  $d$  and depends on the nature of system's boundary conditions (periodic or closed) [40].

To provide more concrete numbers, each tensor network contraction in this paper takes less than 10 s on a laptop computer (Intel Core i5-3360M @ 2.8GHz CPU, 8Gb RAM) for our example system ( $n = 7$  units,  $n_t = 15$  tensors,  $D = 78$ , total number of combinatorial states  $\mathcal{O}(10^{13})$ ). The example system size was chosen to best illustrate the physical phenomena at single unit resolution. In other investigations we reliably contracted lattice networks of up to  $n_t = \mathcal{O}(10^3)$  tensors accounting for  $\mathcal{O}(10^{167})$  combinatorial states using PyTNR [45]. This result suggests that the current tensor network methods would remain tractable for systems even one order of magnitude larger, or perhaps even larger as the tensor network methods develop.



## Appendix B. Supplementary results

### B.1. Avoidance: excess density

In the main text of the paper we show that void premium quantifies the cost of reserving space within the ship hull, with the effect being further amplified by the presence of an anchor (figures 4 and 5). We associated high void premium with a large rearrangement of the functional units. We can quantify the degree of rearrangement by computing the unit density profile without void  $\rho_{\text{no void}}(\vec{x})$ , the unit density profile with void  $\rho_{\text{void}}(\vec{x})$ , and their difference:

$$\Delta\rho(\vec{x}) = \rho_{\text{void}}(\vec{x}) - \rho_{\text{no void}}(\vec{x}), \quad (\text{B1})$$

which we term *excess density* that can be both positive and negative. Since the total number of functional units does not change upon addition of void, the positive and negative regions of excess density have to cancel each other. In this case large rearrangements of the unit cloud are characterized by large *contrast* between the positive and negative regions, graphically visible in saturation of colors.

We plot all three densities in figure 11, both without and with an anchor on unit 1. In low-cost case  $T = 1.0$  (panel (a)) the units want to form a compact cloud, which can be located anywhere within the hull. Upon creation of a void slightly to the left of center, the unit cloud relocates to the right of the void, as seen by large negative  $\Delta\rho$  in the left half of the hull and positive in the right half (visible as red and blue clouds). When unit 1 is anchored (panel (d)), this effect becomes even more pronounced since the unit cloud condenses around a reference point. Creation of a void pushes the unit cloud to the right and above the anchor, but it cannot move

far from the anchor, resulting in high contrast of excess density (strong color saturation in the figure) and thus high void premium. At intermediate and high values of  $T$ , both with and without an anchor (panels (b), (c), (e), (f)) the rearrangements are much smaller, visible in much paler colors on excess density heatmaps.

## B.2. Adjacency: all bond diagrams

In the main text of the paper we show how to compute the bond diagram  $p_{i \rightarrow j}(\theta)$  for any pair of units  $i, j$ . Since the directions  $\theta(i \rightarrow j)$  and  $\theta(j \rightarrow i)$  only differ by a trivial rotation by angle  $\pi$ , and the direction from a node to itself is not defined, a system of  $n$  units would have  $n(n-1)/2$  independent bond diagrams. The logical architecture of the example problem was deliberately chosen to not have any graph symmetries, therefore the bond diagrams are not related to each other via any symmetries.

While in the main text we only show several representative bond diagrams (figures 6 and 7), figure 12 shows all of the bond diagrams for the low-cost regime  $T = 1.0$  and the high-flexibility regime  $T = 3.0$ . The diagrams are arranged as a lower-triangular and an upper-triangular matrix of polar plots so that the diagrams in positions symmetric with respect to the diagonal refer to the same pair of functional units and are thus directly comparable. The full set of diagrams shows all adjacency features highlighted in the main text. For units that are directly connected, most of the diagrams at  $T = 1.0$  show eightfold signal (e.g.  $2 \rightarrow 6$ ), while diagrams at  $T = 3.0$  show X-shaped fourfold signal (e.g.  $4 \rightarrow 5$ ). All diagrams to or from unit 1 show further symmetry breaking because unit 1 is anchored in space. For units that are connected only indirectly with large topological distance, the bond diagram approaches the null model at both  $T = 1.0$  and  $T = 3.0$  (e.g.  $3 \rightarrow 4$ ).

## ORCID iDs

Andrei A Klishin  <https://orcid.org/0000-0002-5740-8520>

Greg van Anders  <https://orcid.org/0000-0002-9746-2484>

## References

- [1] Aikens C H 1985 *Eur. J. Oper. Res.* **22** 263
- [2] Dorneich M C and Sahinidis N V 1995 *Eng. Optim.* **25** 131
- [3] Drira A, Pierreval H and Hajri-Gabouj S 2007 *Annu. Rev. Control* **31** 255
- [4] Owen S H and Daskin M S 1998 *Eur. J. Oper. Res.* **111** 423
- [5] Shields C P F, Rigterink D T and Singer D J 2017 *Ocean Eng.* **135** 236
- [6] Dorsey J W 2003 *Environ. Pract.* **5** 69
- [7] Adams D and Watkins C 2008 *Greenfields, Brownfields and Housing Development* (New York: Wiley)
- [8] Hopkins R and Jenkins K 2008 *Eating the IT Elephant: Moving from Greenfield Development to Brownfield* (Reading, MA: Addison-Wesley)
- [9] Chalfant J 2015 *Proc. IEEE* **103** 2252
- [10] Brefort D et al 2018 *Ocean Eng.* **147** 375
- [11] Newman M 2010 *Networks: An Introduction* (Oxford: Oxford University Press)
- [12] Buldyrev S V, Parshani R, Paul G, Stanley H E and Havlin S 2010 *Nature* **464** 1025
- [13] Li W, Bashan A, Buldyrev S V, Stanley H E and Havlin S 2012 *Phys. Rev. Lett.* **108** 228702
- [14] Shields C P F, Sypniewski M J and Singer D J 2016 *Proc. of the 13th Int. Symp. on Practical Design of Ships and Other Floating Structures (PRADS' 2016)* ed U Nielsen and J Jensen (Technical University of Denmark)
- [15] Newman M E J 2003 *SIAM Rev.* **45** 167
- [16] Shields C P and Singer D J 2017 *Nav. Eng. J.* **129** 75–86 <https://www.ingentaconnect.com/content/asne/nej/2017/00000129/00000004/art00020#expand/collapse>
- [17] Shields C P F 2017 Investigating emergent design failures using a knowledge-action-decision framework *PhD Thesis* (University of Michigan)
- [18] Gibbs J W 1902 *Elementary Principles in Statistical Mechanics* (New York: Charles Scribner's Sons)
- [19] Klishin A A, Shields C P F, Singer D J and van Anders G 2018 *New J. Phys.* **20** 103038
- [20] Jaynes E T 1957 *Phys. Rev.* **106** 620
- [21] Klishin A A, Kirkley A, Singer D J and van Anders G 2020 *Sci. Rep.* **10** 14334
- [22] Penrose R 1971 *Combinatorial Mathematics and its Applications* ed D J A Welsh (New York: Academic) p 221
- [23] Verstraete F, Murg V and Cirac J I 2008 *Adv. Phys.* **57** 143
- [24] Schollwöck U 2011 *Ann. Phys., NY* **326** 96
- [25] Orús R 2014 *Ann. Phys., NY* **349** 117
- [26] Chan G K-L and Sharma S 2011 *Annu. Rev. Phys. Chem.* **62** 465
- [27] Xie Z, Jiang H, Chen Q, Weng Z and Xiang T 2009 *Phys. Rev. Lett.* **103** 160601
- [28] Levin M and Nave C P 2007 *Phys. Rev. Lett.* **99** 120601
- [29] Biamonte J D, Morton J and Turner J 2015 *J. Stat. Phys.* **160** 1389
- [30] Kourtis S, Chamon C, Mucciolo E R and Ruckenstein A E 2019 *SciPost Phys.* **7** 060
- [31] Cichocki A et al 2016 *FNT Mach. Learn.* **9** 249
- [32] Cichocki A et al 2017 *Found. Trends Mach. Learn.* **9** 431
- [33] Oseledets I V 2011 *SIAM J. Sci. Comput.* **33** 2295
- [34] Stoudenmire E and Schwab D J 2016 *Advances in Neural Information Processing Systems* (Red Hook, NY: Curran Associates Inc.) pp 4799–807

- [35] Goldenfeld N 1992 *Lectures on Phase Transitions and the Renormalization Group* (Reading, MA: Addison-Wesley)
- [36] Steinhardt P J, Nelson D R and Ronchetti M 1983 *Phys. Rev. B* **28** 784
- [37] Roth J and Denton A R 2000 *Phys. Rev. E* **61** 6845
- [38] Thouless D J 1974 *Phys. Rep.* **13** 93
- [39] Filoche M and Mayboroda S 2009 *Phys. Rev. Lett.* **103** 254301
- [40] Jermyn A S 2020 *SciPost Phys.* **8** 5
- [41] Rosenblatt M 1956 *Ann. Math. Stat.* **27** 832
- [42] Parzen E 1962 *Ann. Math. Stat.* **33** 1065
- [43] Pfeifer R N, Haegeman J and Verstraete F 2014 *Phys. Rev. E* **90** 033315
- [44] Jermyn A S 2019 *J. Comput. Phys.* **377** 142
- [45] Klishin A A and van Anders G 2020 *Soft Matter* **16** 6523