骨架耦合：时序网络中社区演化的一种新颖层间映射

Bengier ¨Ulgen Kilic[1], *和 Sarah Feldt Muldoon[1], 2,3, †[1]数学系，纽约州立大学布法罗分校，美国纽约[2]*CDSE*项目，纽约州立大学布法罗分校，美国纽约[3]神经科学项目，纽约州立大学布法罗分校，美国纽约
（日期：2023年9月4日）

摘要

时序网络中的动态社区检测（DCD）是一项复杂的任务，需要选择合适的方法及其相关超参数。如何选择最合适的方法，通常取决于所分析网络的类型以及定义该网络的数据的具体特征。对于由神经元尖峰序列数据构建的功能性时序网络，社区通常是瞬态的，而且网络中包含多个单例社区也很常见。在此，我们比较了不同动态社区检测方法在由具有已知社区结构的合成神经元时间序列数据构建的功能性时序网络上的性能。我们发现，对于这类网络，利用层间连接在各层之间实现社区延续的动态社区检测方法，性能优于其他方法。然而，我们也观察到，DCD 性能高度依赖于层间连接的拓扑结构，尤其是在存在单例和瞬态社区时。因此，我们提出了一种在时序网络中定义层间连接的新方法，称为骨架耦合。该方法专为增强网络中社区随时间的连接而设计，其依据是社区历史的拓扑属性。我们表明，将骨架耦合与现有动态社区检测方法相结合，可以提升这些方法在植入单例与瞬态社区的合成数据上的性能。因此，在真实世界神经元数据或其他具有瞬态结构和单例社区的系统中，使用骨架耦合进行动态社区检测，将有助于更准确、易于解释地呈现社区演化结果。

arXiv:2301.10860v2 [physics.data-an] 1 Sep 2023

* bengieru@buffalo.edu
† smuldoon@buffalo.edu

I. 引言

复杂系统通常由一些元素构成，这些元素的动态特性及相互作用会随时间变化。这类时间事件可能描述人类通信 [1], 邻近关系 [2–4], 贸易和交通运输 [5, 6], 引用和合作[7, 8], 或生物 [9, 10] 及神经元相互作用 [11, 12]。将这些系统建模为时序网络 [13, 14] 可能很有用，因为网络节点和边能够捕捉数据的时间属性。这对于节点动态特性可用时间序列数据表示的系统尤为相关。

神经元系统是动态系统的典型示例，可以将其建模为时序网络。例如，脉冲序列数据描述了神经元随时间变化的同步放电模式。因此，我们可以构建一个功能网络，其中节点是神经元，边表示神经元放电模式之间的统计关系（例如同步或其他相似性度量）。为了捕捉神经元对之间的相互作用会随时间变化这一事实，利用此类数据构建时序网络的一种常见方法，是创建网络随时间变化的连续快照，以描述数据的动态演化。为此，可以将时间序列划分为更短的时间序列，构建按时间顺序排列的网络状态有序集，并尝试刻画这些单个快照中的内在连通模式（图1A）。

时序网络中一个常被关注的方面，是网络内社区随时间变化的动态特性（即社区如何随时间产生或消亡，以及随着网络结构演化，节点的社区成员关系如何变化）。在神经元放电的例子中，社区可以代表同步放电的神经元群组（细胞组件），我们可以研究这些群组的成员关系如何随时间变化。已有多种动态社区检测（DCD）方法，将静态社区检测扩展到时序网络，使社区能够跨时间存在（并产生或消亡）[15]。然而，与静态网络的情况类似，每种DCD方法都基于不同的定义，规定如何在网络中检测社区。此外，DCD方法还必须定义如何在各个快照（网络的层）之间延续或分配社区标签。因此，文献中的DCD方法因其处理快照及其时间依赖关系的方式不同而存在很大差异 [16]。有些方法分别处理单个快照，有些方法可能按时间顺序迭代处理快照，还有些方法可能利用层间边将随时间变化的快照连接成一个时间上连通的网络。

在此，我们重点关注五种常用的动态社区检测方法，它们涵盖了定义动态社区的不同方式：多层模块度最大化（MMM） [17], Infomap [18, 19], 动态随机块模型（DSBM） [20, 21], 动态团簇传播方法（DPPM） [22], 和张量分解 [23]。MMM 和 DSBM 根据零模型，将社区定义为由节点密集连接而成的簇；Infomap 则将社区定义为信息流动快速且高效的一组节点。DPPM 采用另一种定义：社区由完全连接子图（团）的子集（团）构成，且这些子集之间的重叠最大。最后，张量分解采用线性代数方法，将社区定义为生成底层网络的向量空间的基。

动态社区定义上的这种差异，迫使这些方法对如何在快照（层）之间进行社区标签的时序延续作出特定假设（图。1B）。MMM 和 Infomap 等方法认为，时序延续是通过结构多层网络拓扑实现的；在这种情况下，会定义连接不同层中节点的“层间边”，使社区能够自然地跨越不同时间存在。然而，另外三种方法采用“固定规则”来定义时序延续，而忽略了数据特有的差异。DSBM 使用固定的时序网络生成模型，通过贝叶斯算法在不同时刻创建并传递社区。DPPM 使用固定算法：如果静态层中的团在不同快照之间有足够大的交集，就会将其延续到之后的时间。最后，张量分解采用固定的分解算法（PARAFAC），将三阶张量分解为更简单的矩阵，其中分解的时间分量对应于时序延续。

重要的是，由于各方法在静态和动态情况下对社区的定义方式各不相同，不同的方法会突出数据的不同特征，因此也会检测到不同的动态社区模式。因此，我们必须了解每种方法如何检测数据的具体特征，并将这些信息纳入检测到的社区中。当这些方法应用于底层真实值未知的实验数据集时，这一点对于解读结果尤为重要。受神经科学中的示例启发，我们尤其关注各种动态社区检测方法在检测单例社区（独立放电的神经元）和瞬态社区（细胞组件随大脑状态变化而改变）的高占比数据时的表现。因此，我们模拟具有已知社区结构的脉冲序列数据，并在这些数据上测试动态社区检测方法的性能。

正如预期，我们发现，对于同一数据集，不同方法检测到的动态社区结构模式各不相同。在我们的模拟数据中，采用层间边连接不同时刻快照的方法，更善于检测单例社区和瞬态社区；但所有方法在实现社区标签的时序延续方面都表现不佳。我们发现，在这些时序网络中定义层间连接的拓扑结构会极大地影响方法的性能。最常见的层间耦合技术称为对角耦合，即在连续层中将网络节点与其自身相连；在我们的数据中，这种方法在实现社区标签的延续分配方面表现不佳。然而，我们发现，利用网络中每个独立层的层内 拓扑结构信息来耦合各层，可以提升

DCD 性能。

借助拓扑数据分析（TDA）这一数学领域，它位于数据科学与代数拓扑的交叉地带 [24–27], 我们提出了一种新颖的层间耦合方法，称为骨架耦合。该方法根据时序网络静态层中的社区信息定义层间边。骨架耦合会考虑顶点的时序邻域历史，以及其社区分配（在相邻过去状态中），从而提升动态社区检测方法在单例社区和较大群组的时序延续方面的性能。我们将骨架耦合的结果与此前提出的层间耦合机制进行比较，并表明，在单例社区和瞬态社区占比较高的数据上，骨架耦合优于其他方法。

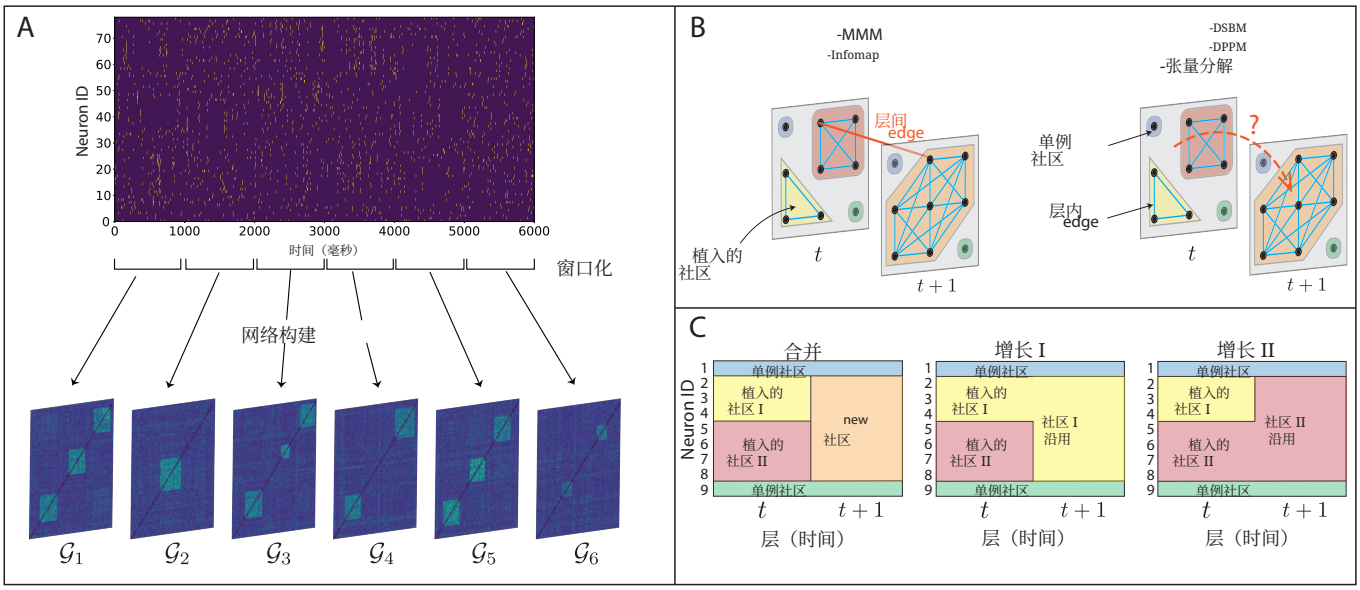

图 1. **从时间序列到动态社区分析 A.** 通过齐次泊松过程生成的包含 $N = 78$ 个神经元的合成时间序列，其中植入的社区每隔 $\tau = 1000$ 毫秒经历一次社区事件。数据被划分为长度为 1000 毫秒的时间窗，并通过计算脉冲序列两两之间的最大互相关，构建出六个代表神经元共同活动的功能网络快照。**B.** 一个包含两个快照的动态网络，其中时间延续通过 MMM 和 Infomap 采用层间边实现（左），或通过 DSBM、DPPM 和张量分解采用某种固定规则实现（右）。**C.** B 部分中发生的社区事件的三种不同情形。左侧，t 中的两个预置社区“合并”，因此形成的社区在 $t+1$ 中具有新的社区标签。中间，植入社区 I 通过与植入社区 II 合并而“扩张”，形成的社区沿用 I 的标签。右侧，植入社区 II 通过与植入社区 I 合并而“扩张”，形成的社区获得标签 II。不同的动态社区检测方法对这些类型的时间延续采取不同的处理方式。

II. 神经元数据的模拟

如前所述，本研究旨在应用于对由单个神经元放电模式构建的时间功能网络进行研究。在这些功能网络中，节点是神经元，连接则通过计算成对神经元放电模式之间的相关性来确定。因此，在此类网络数据中，我们预计会看到许多单例社区（放电模式与其他神经元不相关的神经元），以及多个瞬态社区（不同的同步细胞集群，它们随着神经元短暂地协同放电以执行计算而形成并消亡）。然而，由于真实世界神经元数据中的社区演化真实值无法确定，我们在此将分析应用于合成数据。尽管此前研究社区演化的工作设计了用于测试演化网络中社区检测的基准网络 [28–31], ，但这些网络中的连接代表节点之间的结构连接。因此，这些基准模型通常不包含单例社区或高度瞬态的社区，而这些社区在基于相关性数据的功能网络中很常见 [32–34] ，例如功能神经元网络。为此，我们设计了一组功能基准网络，其基于模拟神经元脉冲序列之间的相关性构建，采用的过程与 [35]中所述的方法类似。有关如何模拟脉冲序列时间序列并计算其相关性的详情，请参见方法部分。

在数值实验中，我们研究动态功能网络中预计会出现的两种不同类型的社区事件：单调事件和非单调事件。单调事件对应这样一种情景：图随时间逐渐演化，使得一层中的社区嵌套于相邻层的社区中。（请回想，在

动态社区划分中，社区由节点—层元素构成，因为节点可以在不同层之间改变所属社区。）在数学上，如果时序网络中所有社区和时间步均满足 p 在时间步 t， C_p^t满足 $C_p^t \subseteq C_p^{t+1}$（或 $C_p^{t+1} \subseteq C_p^t$ 在时间反向时满足），我们就称该社区演化情景为单调情景。此类事件包括社区扩张、收缩或延续。非单调事件指相邻层中的社区可能部分重叠的情景（如图1A 所示），但这些社区不一定彼此包含。非单调事件的例子包括社区合并、分裂、消亡和诞生。在这些情景中，都有必要确定社区标签应如何随时间演化；根据数据的属性（例如神经元放电模式或速率），人们可能希望定义一个新社区，或者沿用先前的标签（见图 1B-C）。

这里，我们聚焦于单调事件（扩张社区）和非单调事件（多个瞬态社区）的示例。社区结构通过模拟神经元尖峰活动来建模，其中预先设定了同一社区内各个神经元放电模式之间的相关性。此外，社区结构（神经元的相关放电）可通过一系列社区事件动态演变。重要的是，在这些数据中，多个神经元的放电模式彼此独立，因此数据中存在许多单例社区。

为了将这些数据映射为时序网络，首先将时间序列划分为多个时间窗，每个时间窗代表网络的一层。通过计算时间窗内神经元放电模式之间两两最大互相关的绝对值，得到每一层的功能网络结构。由于使用互相关来定义功能网络连接会形成一个完全连接的网络，其中包含许多较小的边权值，这些权值很可能代表数据中的噪声，因此对于每个数据集，我们都创建一组时序网络，并使用阈值剔除边权重低于该阈值的连接。在以下章节中，所有结果均展示了多个阈值下的情况（参数空间图中的 x 轴所示，见图 2、5 和 6）。有关合成数据生成和网络构建的更多细节，请参阅方法。

III. 动态社区检测方法性能比较

我们比较了上述两种不同社区演化情景（扩展型社区和瞬态社区）下5种不同动态社区检测方法（MMM [17], Infomap [36], DSBM [20], DPPM [22]和张量分解 [23]）的性能。我们纳入了各方法特有的一系列超参数（分别为分辨率参数 γ、多层松弛率 ρ、度校正 Δ、k-plex维数 k和输入张量秩 η），并在这5种方法的参数空间图中沿纵轴改变这些参数。需要指出的是，尽管这些超参数在不同方法之间不可比较，但用户在实现算法时必须为每个参数做出选择，因此我们改变这些参数，以展示用户选择所产生的影响。

在图2A和图2B的左侧面板中，我们展示了植入的动态社区的社区演化真实值。随后，我们绘制了描述方法性能的参数空间图，其中性能是归一化互信息（NMI） [37, 38] 相对于真实值的函数（见方法部分“评估划分质量”）。在这些图中，代表各方法最优性能的参数值由绿色矩形围成的区域表示（见方法部分“最优区域”）。该最优状态下的社区演化示例显示在参数空间图下方。

在图 2(A) 中，我们展示了 5 种动态社区检测方法检测扩张社区事件中社区演化的性能。起初，神经元各自属于单例社区（其放电模式与其他神经元不相关），随着时间推移，它们逐渐加入一个不断扩大的相关社群（由一系列单调事件构成）。从归一化互信息参数景观可以看出，各方法检测这种社区演化模式的能力有所不同。下方给出了各方法性能最优时的社区演化示例图及对应的方法超参数。可以观察到，MMM 和 Infomap 方法在检测单例社区和实现时间延续方面表现最佳；在更广泛的方法超参数范围内，这两种方法也能得到最高的归一化互信息值（颜色较深的红色）。不过，MMM 未能检测到扩张社区，而 Infomap 虽然检测到了这个不断扩大的社区，但结果带有一些噪声。另一方面，DSBM 和 DPPM 得到的归一化互信息较低，且检测结果总共只有 2 个社区，因为它们无法区分单例社区，而是将所有不相关的神经元归并为一个社区。张量分解的表现介于这两种极端情况之间：它能分别检测出单独层中的大多数社区，但无法实现时间延续。

接下来，我们比较这些方法在包含瞬态社区和单例社区（非单调事件；图 2B）数据上的表现。同样，我们观察到，按归一化互信息（NMI）衡量，MMM 和 Infomap 的表现最佳；但最优社区划分表明，它们检测到的社区演化模式截然不同。这两种方法都能检测到单例社区，并在这些单例社区之间实现时间延续。此外，MMM 还能检测到一些较大的瞬态社区，但无法在层与层之间对这些瞬态社区实现时间延续。张量分解也能检测到瞬态社区和单例社区，但完全无法实现社区层之间的时间延续。再次，DSBM 和 DPPM 仅检测到两个社区，这与植入结构不符，也体现在它们较低的归一化互信息值上。

值得注意的是，在每个 t　在所研究的情形中，DSBM 和 DPPM 方法无法检测到

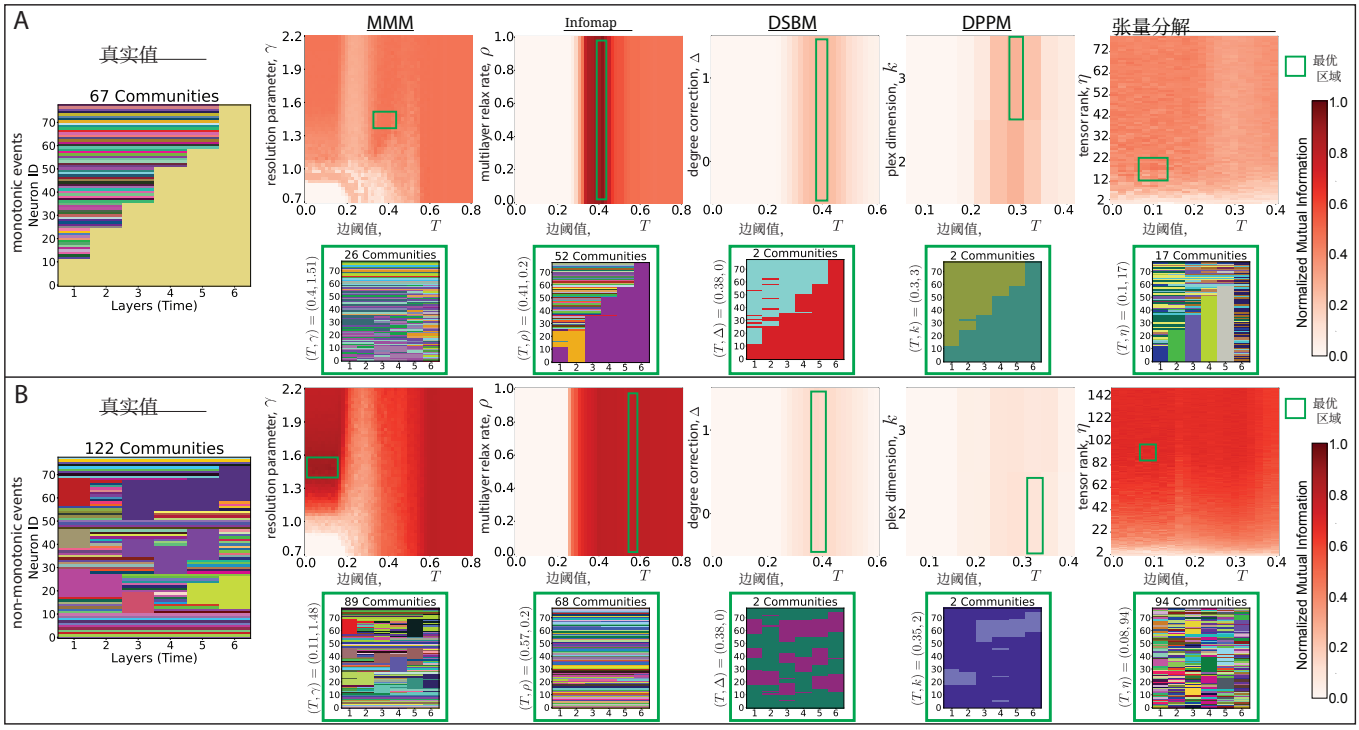

图 2. **在模拟时间序列上比较动态社区检测方法。** 比较五种不同动态社区检测方法（MMM、Infomap、DSBM、DPPM 和张量分解）在参数空间中的性能，并展示其最优区域的示例划分。通过绘制“真实值”社区标签（左侧面板）与预测标签之间的归一化互信息（NMI），考察方法性能随边阈值 T 和特定方法参数（分别为分辨率参数 γ、多层松弛率 ρ、度校正 Δ、k-团簇大小 k 和张量秩 η）的变化情况。最优区域定义为使 NMI 最大化的参数值（T，$\cdot$），并在每个参数网格下方展示该区域的示例划分。划分图展示了六个快照（网络层）中的社区演化，颜色表示每个神经元在各个时间点所属的社区标签。**A.** 模拟 $N = 78$ 个脉冲神经元的动态，使一个大型社区在每一层都通过与单例社区合并而不断扩张（单调事件）。真实值社区演化过程中总共有 67 个社区标签。**B.** 模拟 $N = 78$ 个脉冲神经元的动态，使神经元同步群组随时间出现和消失，即瞬态社区；不属于任何社区的神经元会被赋予唯一的社区标签（以颜色表示），并在时间上延续，除非该神经元经历社区事件。如真实值划分图所示，此过程中共生成 122 个社区标签。

数据中的单例社区。值得注意的是，动态随机块模型无法捕捉单例社区，因为根据这些方法对社区的定义，社区必须是包含多个节点的块结构，而单例社区无法构成块。同样，DPPM 将社区定义为 k-团，即由多个节点组成的集合。因此，单例社区无法构成团。此外，张量分解始终无法实现检测到的社区在时间上的延续，这很可能是该方法中求解器容差造成的近似伪影。虽然 MMM 和 Infomap 并非总能实现时间延续，但它们能够识别单例社区；更重要的是，这些方法依靠层间边来连接不同时间的层。在上文所示的数据中，MMM 和 Infomap 方法采用了对角耦合这一标准技术，因为这是最常用的耦合技术。不过，使用这些方法时还可以调整这一参数；在本文余下部分，我们将重点探讨采用不同的层间耦合技术，以进一步提升 MMM 和 Infomap 方法的社区检测效果。

IV. 传统层间耦合

在时序网络中使用 MMM 和 Infomap 方法时，必须通过层间边构建网络，以描述节点如何跨层（因此也跨时间）相连。本文回顾了用于定义时序网络中层间连接的一些标准技术和近期提出的技术。设 $\mathrm{T} = (V, E)$ 为一个节点对齐时序网络，其快照表示为 $\mathrm{T} = \{\mathrm{G}^1, \mathrm{G}^2, \ldots\ldots \mathrm{G}^{t_{max}}\}$。我们将层 G^t 中的节点 i 记为 i^t，并将从 i^t 到另一个节点 j^s 的边记为 (i^t, j^s)。请注意，在本文余下部分，我们将只关注序数耦合（即节点只允许与相邻层中的其他

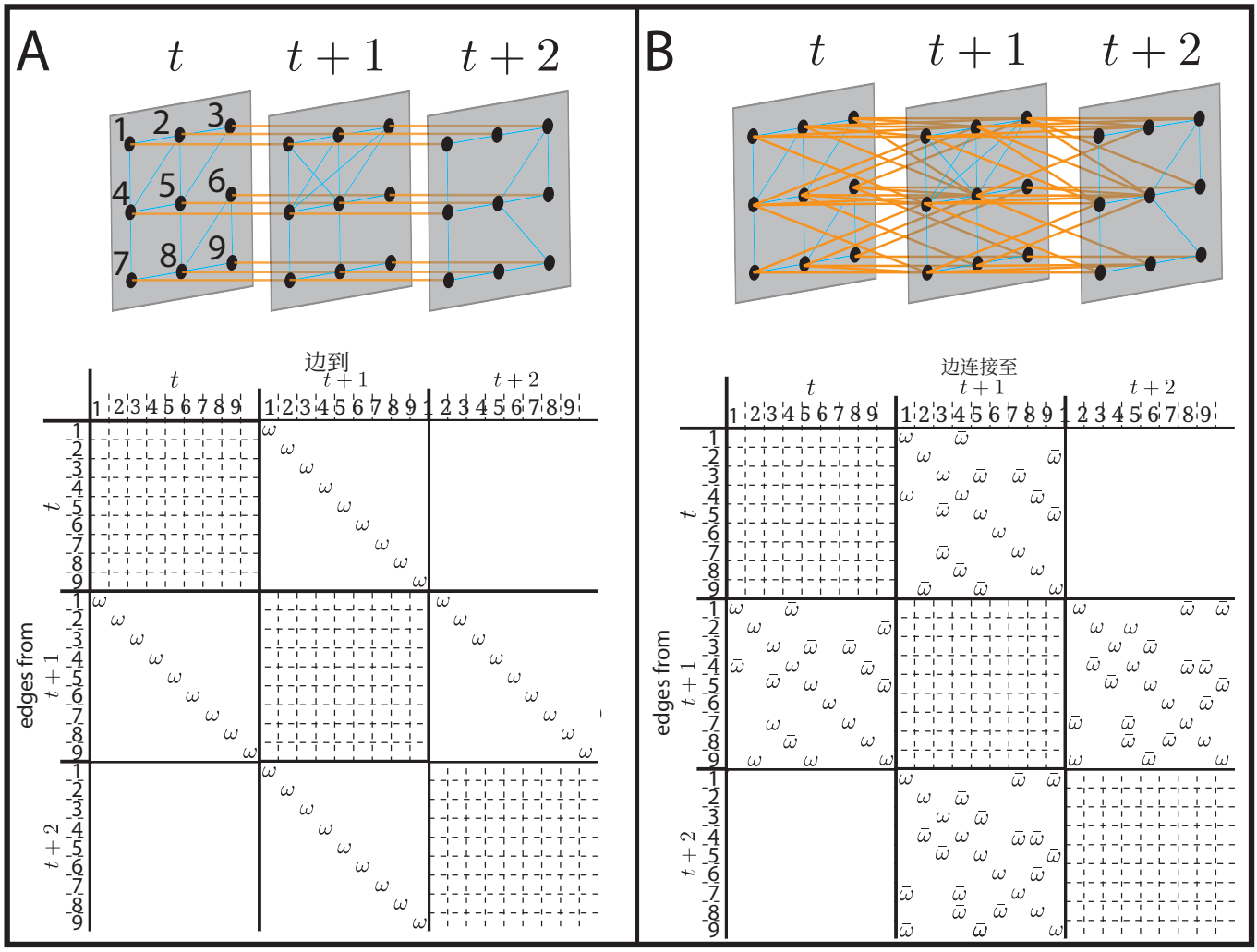

图 3. **层间耦合启发式方法的类型。A.** 对角耦合。节点仅允许以均匀边权重 ω（均匀耦合）连接到自身过去和未来的状态。对此的一种改进（局部更新）是根据节点从时间 t 到 $t+1$的自相似性度量，动态调整边权重。**B.** 非对角耦合。任意一对节点之间都可以存在层间耦合。在此示例中，我们展示了邻域耦合：它将每个节点的邻域与相邻层连接起来，边权重为 ω 和 $\bar{\omega}$。详情请参见方法部分“层间耦合”。

节点相连）。

A. 对角耦合

构建层间边最常见的方法是采用对角耦合。在这种情况下，网络中的每个节点都会按照常规方式与其时间对应节点耦合，如图3A 所示。因此，对所有$(i^t,\ i^{t+1})$ 以及所有$i \in V$，都存在一条 $t \in \{1, 2, .., t_{max} - 1\}$ 层间边，其边权重为 ω，其中 ω 在所有边上均为常数，但其取值必须由用户指定。这里，ω 可以视为一条自身份链接，用于在整个时间过程中保持节点的身份。我们将这种耦合技术称为均匀对角耦合。

虽然这种耦合技术能够让节点在整个网络中保持其身份，但它无法体现这样一个事实：在许多网络中，节点代表的是属性会随时间变化的动态实体。我们可能会问，改变层间边权重的值（即允许 ω 随节点和层而变化）是否会对动态社区的检测产生影响。每条对角层间边都是从节点指向其未来或过去自身的连接，因此，从这个意义上说，这些连接表示节点时间自相似性的强弱。在我们的模拟数据中，节点实际上是神经元，其放电率和模式会发生变化，因此节点的自相似性不一定会随时间保持恒定。先前的研究 [35] 介绍了一种技术，可根据节点 ω 的水平改变

自相似性。在快照之间，节点自相似性的变化越大（例如神经元的放电率变化越大），相应时间层之间的节点层间边权重就越弱。我们将这种技术称为**带局部更新的对角耦合**；在数学上，对于所有 $i \in V$，以及所有 $t \in \{1, 2, .., t_{max} - 1\}$，我们在 i^t 与 i^{t+1} 之间赋予一条层间边，其边权重 ω_i^t取决于节点 i从 G^t 到 G^{t+1}的脉冲率变化。详情请参见方法部分中的“层间耦合”。

B. 非对角耦合

虽然对角耦合只允许网络各层中某个节点与其自身之间建立连接，但放宽这一限制、允许部分或全部节点对之间建立连接同样完全合理；这会引入新的复杂性维度，并使参数空间大幅扩大。尽管可以通过多种方式实施非对角耦合方案，但此处我们重点介绍一种此前提出的算法，称为 **邻域耦合** [39]，，该算法将每个节点周围的最大邻域与相邻层连接起来（图3B）。在数学上，我们为层间边赋予恒定权重 ω，从 i^t 到集合 $\{j^{t+1}\}_{j \in N_i^t}$，使得 j 根据边权重位于 i^t 的最大邻域中（即 i 在 G^t中的强连接邻居），其中 $N_i^t = \{j^t | (j^t, i^t) \in E_t\}$。详情请参见方法部分“层间耦合”。这种耦合算法基于以下假设：网络在先前状态下的拓扑结构会影响网络的演化。请注意，如图 3B 所示，该算法产生的耦合也比对角耦合更稠密。重要的是，尽管该方法基于层 t 中节点 i的层内拓扑结构创建层间边，但这种拓扑结构不一定会在层间拓扑结构中编码 (i,t) 的社区分配，因为节点可能与不同社区中的其他节点相连。无论如何，先前研究表明，使用邻域耦合有助于检测瞬态社区 [39]。

V. 骨架耦合

虽然邻域耦合的优点是能够将网络当前状态的拓扑结构纳入层与层之间的耦合，但这种方法并未直接解决我们希望改善层间社区延续的问题，因为节点的邻域不同于该层中的社区分配。因此，我们提出一种新的非对角耦合算法，称为**骨架耦合**，旨在根据各层内部的静态社区结构连接网络层，从而促进社区标签在时间上的正确延续。

骨架耦合的主要思想是根据节点的时序邻域历史，稀疏或密集地为节点分配层间连接，从而在快照之间创建时间通道。此外，根据动态社区的定义，层间耦合连接只能存在于时间上延续的社区之间，因为这些连接是有向映射（这是由时间的不对称性决定的），将系统的过去状态映射到未来状态。因此，我们首先在每个快照（网络层）中寻找静态社区，以确定这些映射的定义域和值域。也就是说，我们找出时序网络 $P_t = \{C_1^t, C_2^t, C_3^t, ...\}$ 的所有快照 G^t 的划分，其中 T，其中 $C_p^t = \{i^t | \exists i \in V\}$ 由属于社区 p 的网络节点组成，该社区由该层上的静态社区检测方法得到 t。这一步可以使用任何静态社区检测方法完成，不过这里我们使用了稍后采用的动态社区检测（DCD）方法的静态版本（图4A）。

接下来，我们将定义社区的 k-骨架。首先，我们将 k-单纯形 (0, 1, ..., k) 定义为由基数为 $k+1$ 的顶点组成的有序集，例如，0-单纯形是一个顶点 (i)（或简称 i），而 1-单纯形是一条无向无权边 (i, j)。我们将社区 C_p^t 的 K 维单纯复形定义为顶点集 C_p^t 上所有维度不超过 K 的可能单纯形的并集，即 $\{SC_{C_p^t}\}_{k=0}^K$。最后，社区单纯复形的 k 维骨架（或简称社区的 k-骨架）是顶点集 C_p^t 上所有维度小于或等于 k 的单纯形的并集，即 $\{SC_{C_p^t}\}^k$。图4B 底部给出了一些示例。

为简化计算，我们重点考虑 $k \leq 1$ 的情况。需要注意的是，社区至少需要 $k+1$ 个元素才能形成 k-骨架。例如，当 $k = 1$时，$|C_p^t|$ 必须大于或等于 2。因此，单例社区，即 $|C_p^t| = 1$，无法形成 1-骨架。骨架耦合算法的关键在于，1-骨架和 0-骨架在拓扑上是不同的对象（因为前者包含边，而后者不包含），因此我们会相应地决定是否在这些对象之间跨层分配层间连接。

特别地，对于层 G^t 中的任意节点 i^t，有两种可能。第一种是 i^t 可以属于一个大小大于或等于 2 的社区，即满足 $i^t \in C_p^t$的$|C_p^t| \geq 2$。在这种情况下，i^t 是 1-骨架 $\{SC_{C_p^t}\}^1$）的一部分。第二种是 i^t 可以是一个单例社区，即满足$\{i^t\} = C_p^t$ 的$|C_p^t| = 1$。在这种情况下，i^t 具有 0-骨架 $\{SC_{\{i^t\}}\}^0$）。接下来，我们考察 i^t在层 G^{t+1} 中的对应节点，以确定 i^{t+1} 是否属于 0-骨架或 1-骨架。为了确定如何设计层间耦合，我们接着考虑 i^t 和 i^{t+1}的骨架之间所有 4 种组合可能：

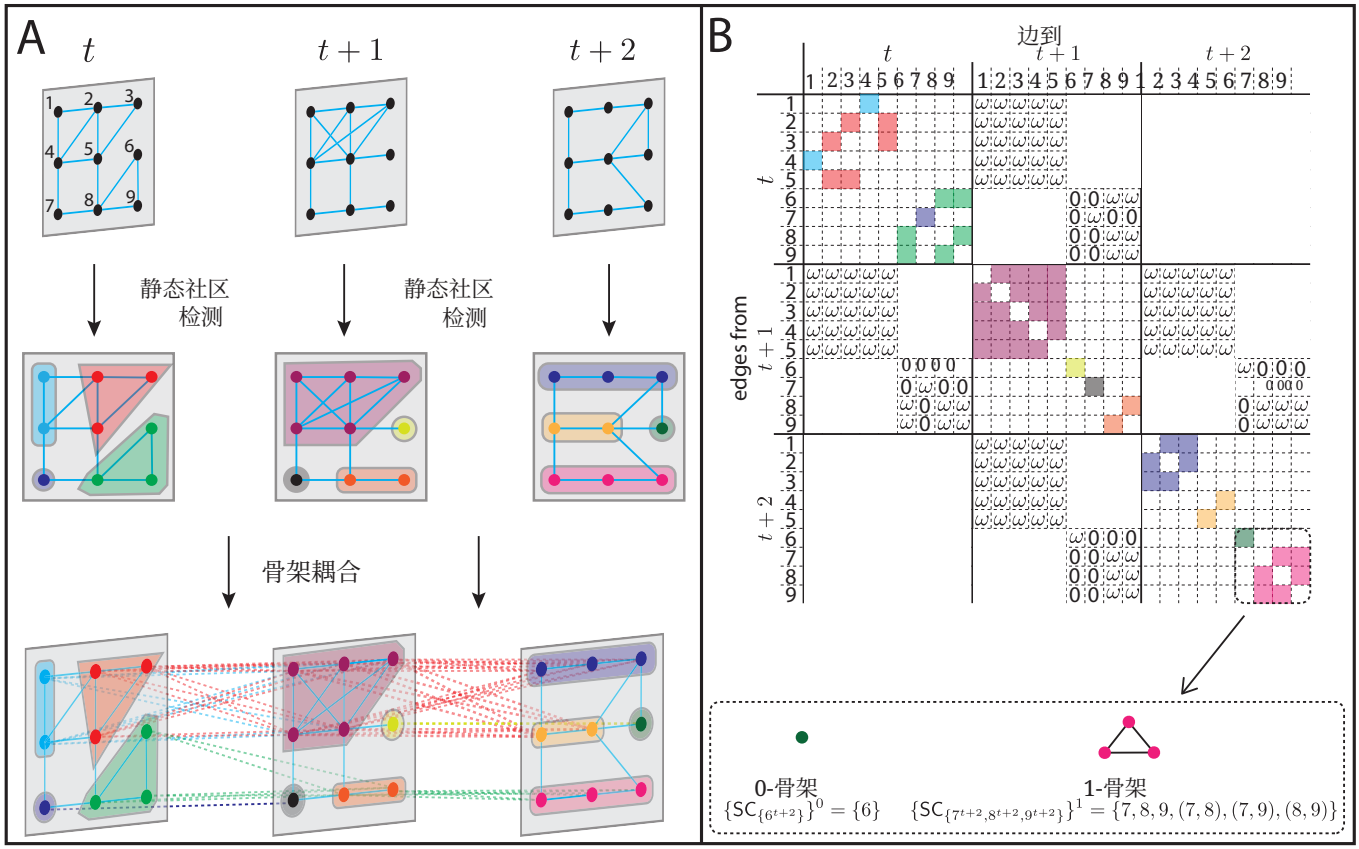

图 4. **骨架耦合。A.** 我们提出的骨架耦合框架示意图，用于在动态社区检测中确定非经验层间边。对单独层执行静态社区检测，以找出每一层中的社区（以颜色表示）。随后通过骨架耦合指定层间边，并对所得时序网络应用动态社区检测。**B. A.**中玩具示例的超邻接矩阵。颜色表示由初级静态社区检测所发现的相应社区中的节点构成的 0-骨架或 1-骨架。例如，条目 $\mathsf{G}^{t+2}{}_{(6,\ 6)}$ 的颜色表示顶点集 $\{6^{t+2}\}$ 上的 0-骨架。或者，条目 $\mathsf{G}^{t+2}{}_{(7,\ 8)}$、$\mathsf{G}^{t+2}{}_{(8,\ 7)}$、$\mathsf{G}^{t+2}{}_{(7,\ 9)}$、$\mathsf{G}^{t+2}{}_{(9,\ 7)}$、$\mathsf{G}^{t+2}{}_{(8,\ 9)}$ 和 $\mathsf{G}^{t+2}{}_{(9,\ 8)}$ 的颜色表示顶点集 $\{7^{t+2}, 8^{t+2}, 9^{t+2}\}$ 上的 1-骨架。我们还在面板底部展示了这些骨架作为单纯复形绘制的示例及相应的数学描述。请注意，骨架耦合是一种均匀、序数且非对角的耦合。为确定层 G^t 和 G^{t+1}之间的层间边，我们比较顶点所构成的骨架。关于层间边的描述，请参阅正文。

：情况 i： i^t 和 i^{t+1} 均为 0-骨架：我们赋予一条无向层间边（i^t, i^{t+1}），其边权重均匀分布为 ω（即对 0-骨架进行对角耦合）。通常，这种情况描述的是由同一个单例随时间维持的社区标签的延续。图 4B 展示了这一情况，其中 7^t 和 7^{t+1} 均为 0-骨架（$\{\mathsf{SC}_{\{7^t\}}\}^0$ 和$\{\mathsf{SC}_{\{7^{t+1}\}}\}^0$ 分别由深蓝色和黑色方块表示）——它们之间有一条权重为 ω 的双向自同一链接（$7^t, 7^{t+1}$）。类似地， 6^{t+1} 和 6^{t+2} 也由一条自同一边相连。

：情况 ii： i^t 是一个0-骨架，而 i^{t+1} 属于一个1-骨架：在这种情况下，我们不会为 i^t 分配任何通往下一个快照G^{t+1}的层间边，因为我们不希望单例节点加入更大的社区后，单例社区标签仍然保留。例如，在图4B 中，可以看到 7^{t+1} 是一个0-骨架（$\{\mathsf{SC}_{\{7^{t+1}\}}\}^0$ 以黑色方块标示），但 7^{t+2}属于一个1-骨架（$\{\mathsf{SC}_{C_p^{t+2}}\}^1$，其中 $C_p^{t+2} = \{7, 8, 9\}$ 以粉色方块簇标示）。因此，我们没有在两个快照之间为该节点分配任何相关的层间连接，如超邻接矩阵对应条目中的权重0 所示。

：情况 iii： i^t 属于一个1-骨架，而 i^{t+1} 是一个0-骨架：这种情况是情况 *ii*的时间反转版本。我们不会从节点 i^t 向时间步G^{t+1} 中的任何节点分配层间边，因为这种情况描述了社区收缩和分裂，我们不希望节点 i^t 的社区标签延续到下一个时间步。请注意，例如，在图4B 中， 6^t 属于一个1-骨架（$\{\mathsf{SC}_{C_p^t}\}^1$，其中$C_p^t = \{6, 8, 9\}$ 由绿色方块簇表示），而 6^{t+1} 是一个0-骨架（$\{\mathsf{SC}_{\{6^{t+1}\}}\}^0$ 由黄色方块表示）。因此，我们没有分配任何层间连接

与该节点相关联的层间连接不存在于这两个快照之间，如超邻接矩阵相应条目中的权重 0 所示。

：情形 iv： i^t 和 i^{t+1} 都是 1-骨架的一部分：我们为快照 G^{t+1}中与i^t 共享社区 C_p^t 的每个节点，向与 i^{t+1} 在快照 G^{t+1}中共享社区 C_q^{t+1}的其他所有节点添加强度均匀分布的层间边 ω。根据 i^t 和 i^{t+1} 所属社区的规模，这种情况可能对应多种类型的社区事件。无论如何，我们希望社区标签随时间保持不变。例如，在图4B中，注意到 3^t 和 3^{t+1} 所属社区的规模大于 1 $C_p^t = \{2^t, 3^t, 5^t\}$ 和 $C_q^{t+1} = \{1^{t+1}, 2^{t+1}, 3^{t+1}, 4^{t+1}, 5^{t+1}\}$ 所属社区（相应的单纯复形 $\{\mathrm{SC}_{C_p^t}\}^1$ 和 $\{\mathrm{SC}_{C_q^{t+1}}\}^1$ 分别由红色和紫色方块簇表示）。这意味着我们添加从 3^t 到 C_q^{t+1}的双向层间边，权重为 ω。如果我们查看 C_p^t、 2^t和5^t的其他元素及其在下一层中的对应元素 2^{t+1}和5^{t+1}，会发现类似的情况。因此，我们在社区 C_p^t 的所有节点与社区 C_q^{t+1}的所有节点之间添加权重为 ω的双向连接，在它们之间建立一座时间桥。

因此，骨架耦合是一种经过精细调整的耦合算法，它基于时变网络中单例与较大社区之间的拓扑差异。在这里，我们借用了拓扑数据分析（TDA）中的“k-骨架”这一术语 [24–27] 因为我们认为，功能网络中动态社区的定义形式为 k-团 [22]。在无噪声数据的理想情况下，由 n 个节点组成的社区应为一个 n-团，而在现实中，动态社区则是一组存在缺失连接或含噪连接的节点。因此，我们借助单纯复形中骨架的定义和用法，来处理真实世界数据；在这类数据中，社区内几乎不可能出现真正的团。我们还在蓝色的附录 A“骨架耦合算法”中提供了骨架耦合实现的伪代码。

VI. 骨架耦合在时序网络分析中的应用

在第 III 节中，我们展示了 MMM 和 Infomap 方法能够检测出数据中的单例社区，但在延续正确的社区标签方面表现不佳。在此前的比较中，我们着重考察了分辨率参数 γ（MMM）和多层松弛率 ρ（Infomap）随边阈值 T变化时的影响。在这里，我们将固定 γ和 ρ，转而探究采用四种不同层间耦合策略的效果：均匀对角耦合、带局部更新的对角耦合、邻域耦合，以及我们新提出的骨架耦合。对于每种策略，我们考察 DCD 方法的性能如何随层间边权重 $(T,\ \omega)$和边阈值 T变化；具体而言，我们通过在算法的多次随机运行中，对真实标签与预测标签之间的归一化互信息（$< NMI >$）取平均值来衡量性能。在 NMI 图下方，我们展示了这些随机运行中每一对参数 $\sigma(NMI)$的 NMI 标准差 $(T,\ \omega)$。更多结果见附录 D“随机运行之间的最大重叠距离”。与前文一样，我们比较了两种不同类型的社区演化情景下的方法性能：一种是单调的社区事件序列，即初始社区随时间扩展，直到整个网络实现同步；另一种是非单调事件，其中 瞬态社区会随时间出现或消失。在附录 B“更多社区演化情景”中，我们展示了另外两组社区演化情景的类似结果：保持神经元数量和层数不变，同时改变各层中社区的大小。请注意，我们没有进行改变层数的实验，因为其他先前研究已探讨过层数如何影响 DCD 性能 [36, 40]。

现在展示如何将骨架耦合纳入网络设计，从而提升 MMM 和 Infomap 方法在时间延续方面的性能。

A. 骨架耦合对 MMM 的影响

我们首先比较 MMM 在不同层间耦合策略下的性能。在图 5（A）中，我们展示了针对扩张社区使用均匀对角耦合和带局部更新的对角耦合所得到的结果。在下方$< NMI >$和 σ（NMI）图中，我们展示了通过共识划分获得的社区演化图，该共识划分汇总了所有最优和非最优参数选择下的随机运行结果。 [41]可以观察到，这两种耦合技术产生了结构上相似的结果，未能识别出扩张社区，反而延续了独立社群标签。对于最优和非最优的耦合参数选择，结果均如此。

接下来，我们研究采用邻域耦合和骨架耦合的非对角耦合方法。值得注意的是，这两种非对角方法都成功检测出了扩张的植入动态社区。然而，邻域耦合无法在时间维度上延续单例社区，而是为每一层中的每个节点都分配不同的社区标签，最终得到171个社区。另一方面，与其他方法相比，骨架耦合更好地延续了单例社区，检测到的社区总数也处于真实值的合理范围内。我们

还观察到，在非最优参数区域中，该方法的整体性能有所提升：接近最优的参数区域（黄色方框）能够部分恢复扩张社区。

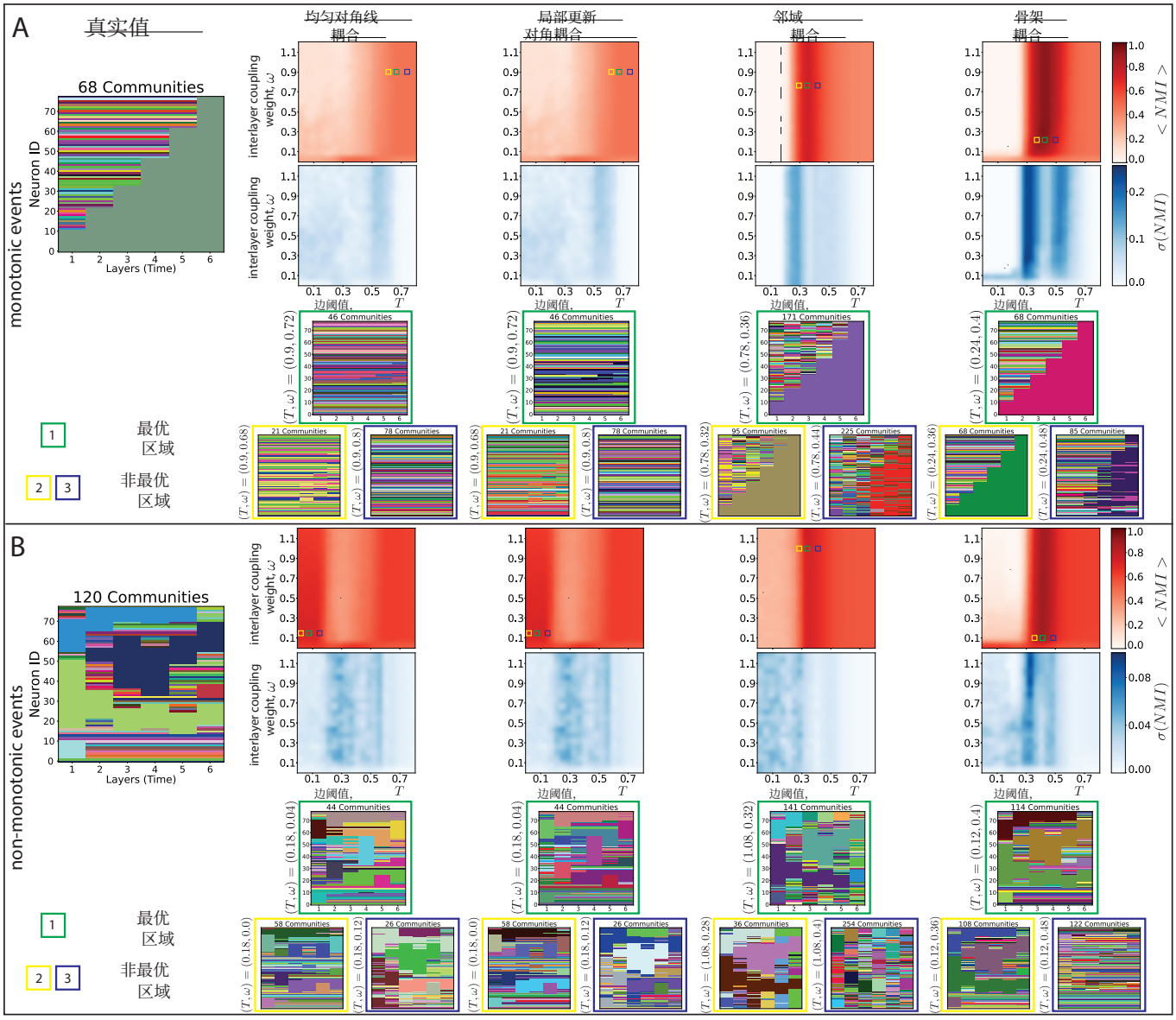

图5. **MMM 的不同层间耦合策略比较**。对于$N = 78$ 神经元经历 **A.** 扩展型社区事件（共 68 个社区标签）和**B.** 瞬态社区事件（共 120 个社区标签），比较不同层间耦合策略。社区标签以颜色表示。参数景观展示了该方法在（T，ω）参数空间中的性能，以多次随机运行的平均 NMI 和标准差衡量。对于 A 中的扩展型事件，采用配置模型作为零模型，分辨率参数设为 0。98；对于 B 中的瞬态事件，分辨率参数设为 1。3。在各参数空间下，我们展示 MMM 方法在绿色、黄色和蓝色矩形标示的最优区域与非最优区域范围内找到的示例共识划分，分别按平均 NMI 从高到低排列。

在图 5(B) 中，我们接下来考察不同耦合策略在检测数据中的瞬态社区时的表现。请注意，两种对角耦合技术在发现层内植入的瞬态社区方面都表现较好。然而，尽管一些单例社区标签会随时间在层之间传递，对角耦合技术在快照之间传递瞬态社区标签方面表现不佳。

使用非对角耦合算法时，我们发现，邻域耦合和骨架耦合在检测并延续瞬态社区标签方面都表现良好。然而，邻域耦合在延续独立社区标签方面表现不佳，因此检测到的社区总数较多。骨架耦合不仅更善于检测并延续瞬态社区，也能更好地延续独立

社区标签，从而取得更高的性能值。总体而言，在与 MMM 结合时，对于两种社区演化情景，骨架耦合的表现均优于其他耦合启发式方法。此外，正如各参数空间下方的标准差图所示，我们认为，这些结果对于不同随机网络生成中的最优划分具有稳健性。可以观察到，对于最优参数选择 $(T\omega)$，不同随机运行间的归一化互信息标准差较低。最后，所有这些结果同样适用于类似的社区演化情景，详见图 B.1。

B. 骨架耦合对 Infomap 的影响

在图 6 中，我们展示了四种不同耦合策略与 Infomap 方法结合时的性能，所用数据与图 5 中的相同。请注意，在图 6（A）中，对于扩张社区，两种对角耦合技术似乎都未能将在时序网络最初几个快照中包含的神经元检测为扩张社区的一部分，这可能是由于时序网络中的快照数量所致。然而，这些耦合方案在捕捉单例社区标签的时间延续性方面表现相当不错。

对于非对角耦合方案，两种非对角方法都以较高的归一化互信息识别出了扩张社区。然而，邻域耦合未能在时间上延续单例社区，为每个节点在每一层都分配了不同的社区标签。相比之下，骨架耦合能够更好地检测单例社区的时间延续性，从而使检测到的社区演化与真实值之间具有相对较好的一致性。

请注意，对于对角耦合方法，$< NMI >$较高（红色较深），而非对角耦合方法则较低。这似乎是对参数空间取平均造成的假象。对角耦合方法与 Infomap 结合时，在相同参数值下，不同随机运行所得的归一化互信息往往更加一致且较高（T，ω）。因此，平均归一化互信息较高。另一方面，非对角耦合方法并非如此，尤其是骨架耦合。不过，共识最优划分表明，与其他任何方法相比，骨架耦合更能确定预置社区结构。

在图 6（B）中，我们探究了各种耦合策略在数据中检测瞬态社区的表现。有趣的是，与 Infomap 方法结合时，对角耦合方法表现很差，无法在各层中保留预置的瞬态社区。相反，使用这些耦合技术往往会使单例社区的社区标签得以保留。另一方面，如归一化互信息值（红色较深）所示，我们再次发现，非对角耦合方法的表现优于对角耦合方法。事实上，示例划分表明，邻域耦合和骨架耦合都能识别植入的瞬态社区。然而，邻域耦合无法在时间上保留单例节点（与图 6（A）类似），导致社区标签总数增加。同样，骨架耦合在时间上保留单例社区的表现更好，并且检测到的社区演化与真实值之间呈现较高相关性。此外，对于具有相同植入动态社区结构的不同网络，在如低标准差所示的最优区域内，这些结果均具有稳健性。最后，对于类似的社区演化情景，以上结果同样成立，相关内容见图 B.2。

VII. 讨论

现实世界的复杂系统呈现动态行为，即系统状态会随时间变化。因此，可以使用时序网络和动态社区检测（DCD）来评估网络社区的演化。在此，我们考察了 5 种不同的动态社区检测方法，并发现这些方法的当前实现无法有效处理包含大量单例社区和瞬态事件的数据。我们还发现，采用层间边耦合策略的方法（MMM 和 Infomap）在识别单例社区方面表现更好。这一观察促使我们将研究重点放在使用层间耦合的动态社区检测方法上。不过，值得注意的是，另外三种未使用层间连接的算法采用规则来确定社区标签如何在各层之间传播。人们或许可以制定其他规则，以改进对单例社区和瞬态社区的检测；未来研究可以进一步探索这一思路。

使用层间连接时，我们进一步发现，采用非对角耦合算法还能更好地实现社区分配的时间延续。因此，我们开发了一种新颖的非对角层间耦合方案，称为骨架耦合。该方案利用相邻先前网络状态中编码的时序邻域历史，以算法方式确定层间边的放置位置。在合成数据中，骨架耦合通过在时间上延续单例社区和大型社区的标签，优于现有的层间耦合方案。

骨架耦合基于这样一种观点：单例社区在拓扑上不同于规模较大的社区。我们将动态社区视为独立于其在网络中连通性的 1-骨架（或划分团）。换言之，给定一个将网络划分为多个社区的划分，某个社区的 1-骨架是由完全连接且不与外部相连的子网络节点构成的，从而将该社区与网络的其余部分离散开来。由于单例社区（即 0-骨架）内部没有任何边，因此不可能拥有 1-骨架，而规模较大的

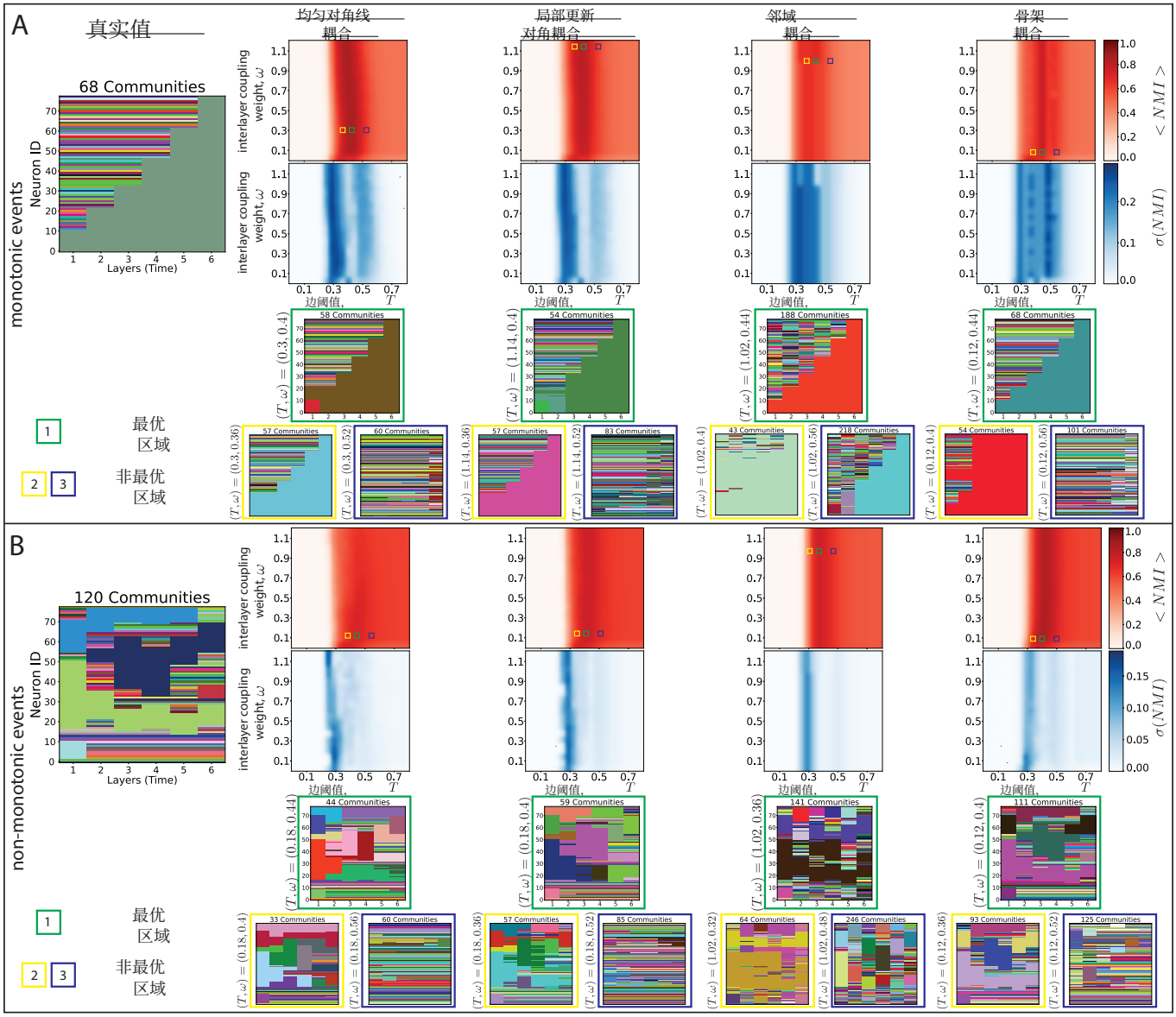

图 6. **Infomap 的不同层间耦合策略比较。** 对经历扩张型社区事件的 $N = 78$ 神经元采用不同层间耦合启发式方法的比较：**A.** 共包含 68 个社区标签的扩展型社区事件；以及**B.** 共包含 120 个社区标签的瞬态社区事件。社区标签以颜色表示。参数景观展示了方法在（T，ω）参数空间中的表现，以平均归一化互信息和标准差衡量。两个面板中的多层松弛率均设为 0。2。在每个参数空间下方，我们展示了 Infomap 方法在绿色、黄色和蓝色矩形标示的最优与非最优区域范围内找到的共识划分示例，分别按平均归一化互信息从高到低排列。

社区中成员之间存在边，因此至少可以拥有 1-骨架。因此，0-骨架和 1-骨架在拓扑上有所不同，必须采用不同的方式进行耦合。通过考察时序网络中社区骨架随时间的演化，骨架耦合算法将时序网络的连通分量连接起来，这对应于在骨架表示中对社区进行离散化的版本之间分配层间边。

新型非对角耦合方案的开发还受到这样一个事实的推动：使用对角耦合方案时，选择合适的边权重、阈值和动态社区检测方法的重要性可能会变得更加复杂。在图5 和6的参数空间图中，我们可以总体观察到，层间边权重的取值似乎不如阈值参数的取值重要，这可从归一化互信息的相似颜色值在图中沿垂直方向延伸看出。事实上，使用MMM方法比较图5A和B时，显然对于对角耦合方案，最优阈值参数高度依赖于社区事件的类型。对于非对角耦合方案，这种效应则明显弱得多。观察图6中Infomap的性能时，我们再次看到类似的效应。

这里，对于对角耦合方案，我们再次看到A和B面板之间最优方法超参数区域存在差异。有趣的是，对于与Infomap结合使用的对角耦合方案，其对边权重选择的依赖性也更强。此外，采用非对角耦合方案（邻域和骨架）时，图5和6 中的最优区域对于扩展型和瞬态社区事件均表现出强得多的归一化互信息，并且其沿整个参数空间（垂直方向）延伸，这一观察结果与动态社区检测方法的选择无关。这一发现表明，骨架耦合等非对角连接方案可用作一种降维技术，因为最优方法超参数的选择对层间耦合边权重的依赖性较低，ω、社区事件类型以及方法选择的影响较小。最后，请注意，图5 和6中所有图的最优区域参数值的标准差都较低，因此这表明我们的结果在网络生成的不同随机运行中具有稳健性。

尽管非对角耦合方案有许多优势，骨架耦合的一个缺点是该框架的计算成本。为了利用骨架耦合，需要对单个快照应用静态社区检测，因为骨架耦合利用静态社区信息来确定层间边。这意味着，在对完整时序网络运行动态社区检测方法之前，需要多次运行动态社区检测方法的静态版本，这显然会增加计算复杂度。然而，对于短堆叠时序网络（快照数量较少），鉴于动态社区检测方法的准确性大幅提升，该方法的耗时并不成问题。

值得注意的是，不同的社区检测方法（包括静态和动态方法）并非确定性的，也就是说，每次运行方法都可能得到不同的划分，尽管这些结果在结构上可能相似。尽管我们对每个合成神经元时间序列多次运行动态社区检测方法，并在全文结果中展示了动态共识划分，但骨架连接是通过同一种静态社区检测方法，在每一层仅运行一次其静态版本来确定的。在我们的案例中，每层只运行一次算法纯粹是出于计算方面的考虑。对每层的检测结果进行共识处理会增加计算复杂度，结合我们所探索的大参数空间，这样做并不现实。在计算复杂度不成问题的情况下，可以通过增加一个步骤来确定骨架连接：对每一层 t、$\{P_t\}_{k=0}^{10}$运行静态社区检测方法的算法，并根据不同运行结果计算静态划分集成，然后确定每一层的共识划分 CP_t。在这种情况下，可以依据该共识划分确定骨架连接。

最后，我们指出，骨架耦合的同一理念还可以进一步扩展到高阶骨架。例如，规模为 2 的社区在拓扑上也不同于规模为 3 及以上的社区，因为规模为 2 的社区至多拥有 1-骨架，而规模更大的社区至少可以拥有 2-骨架，这对应于相应单纯复形中填充的三角形。一般而言，社区规模与其关联骨架的维度相关，规模为 k 的社区至多可以拥有 ($k-1$)-骨架，从而将其与规模更大的社区区分开来。在我们提出的框架中，如果进一步利用高阶骨架来选择层间边，则会在所提出算法的 *'case iv'* 部分中增加若干子情况。我们预计，引入高阶骨架可以通过增大层间拓扑耦合的差异来提升动态社区检测方法的性能。然而，由于前文讨论的计算方面的顾虑，本文仅关注维度不超过 1 的骨架。

总之，本研究通过比较和对照各种动态社区检测方法及其用于时序网络社区检测的最优方法超参数，填补了文献中的一项重要空白。在真实值社区结构及其演化未知的真实世界数据中，了解网络构建方式和动态社区检测方法的选择如何影响检测到的社区演化结果至关重要。因此，对于预期存在单例社区和瞬态社区的数据集，我们建议采用骨架耦合等非对角耦合算法，以提升方法性能，并更准确地呈现社区演化。这对于功能神经元网络尤为重要：由于对神经元群体的欠采样，我们预期存在许多相互独立（不相关）的神经元，从而形成许多单例社区；而细胞组装组成的快速演化则会产生许多瞬态且不断演化的社区。最后，尽管本研究的动机源自神经科学数据构建的网络，但本文所述方法并不局限于此类数据；对于任何可能存在单例社区和／或瞬态社区的时序网络，同样适用这些建议。

VIII. 方法

A. 代码与数据可用性

骨架耦合的实现代码可在以下位置找到： [42]此代码库的配套文档可在以下位置找到： [43]。

B. 动态社区检测方法

为考虑动态社区检测算法中的随机性，本文所呈现的结果中，以下每种动态社区检测方法均运行了 10 次。随后，根据最大重叠共识（MOC）方法计算得出共识划分 [41]，该划分基于所有算法运行结果。请注意，所有划分均表示每一层中各节点的社区分配；同一社区标签可跨层（随时间）存在，且时间网络数据中的社区可能产生或消亡。

1. 多层模块度最大化（*MMM*）

模块度通过比较社区内节点之间及不同社区间节点之间的连通性，并与零模型中的预期情况进行对比，来评估划分质量 [17]。在我们的研究中，层内社区与配置模型进行比较 [44]，并使用Leiden求解器 [45]（而非常用的Louvain算法[46]）。在图2中，我们探究了方法性能随边阈值和分辨率参数变化的情况，（T，γ）。在此分析中，所有计算均假设采用均匀对角耦合，且层间边权重为 $\omega = 0.1$。在图5的分析中，我们选取了图2 的分析所得的最优层间边权重（图5A，$\gamma = 0.94$；图5B，$\gamma = 1.46$），并固定该参数，以探究不同层间耦合配置下的（T，ω）参数空间。

2. Infomap

Infomap 根据随机游走者通过地图方程访问节点的次数来确定社区结构[18, 36]。我们使用 Python API [19]，采用有向流模型并优化两层划分来开展分析。我们探究了方法性能随边阈值和多层松弛率的变化情况，(T，ρ)。在此分析中，所有计算均假设采用均匀对角耦合，且层间边权重为 $\omega = 0.1$。对于图5中的分析，我们选取了图2 （图5A 和 B，$\rho = 0.2$）分析所得的最优多层松弛率，并将该参数固定，以探索不同层间耦合配置下的(T，ω) 参数空间。

3. 动态随机块模型（*DSBM*）

随机块模型通过尝试将生成模型拟合到数据的已知属性来确定社区结构[20, 21]。在我们的实验中，我们使用了 *LayeredBlockState*（来自 graph-tool API） [47]，采用重叠模型，并将边协变量设为“实指数” [20]。我们的分析分别在启用和未启用度校正的情况下进行 Δ (1 和0，依次对应），并在(T，Δ) 参数空间中将其作为图2中的两行。

4. 动态团传播方法（*DPPM*）

动态团传播方法是团渗流方法（CPM）的一种推广 [48]。DPPM 放宽了社区定义中的条件：在 CPM 中，社区被定义为 n-团，而在 DPPM 中则定义为 k-团，作用于 n 节点 [22]。该方法在网络的单独层上运行，以寻找拓扑聚类的团，并用它们定义静态社区。随后，这些社区标签会分别跨快照延续，用于映射与匹配。我们的分析采用了 $n = k + 2$，针对$k = 2$、3，如图(T，k) 所示参数空间中的行所示2。此外，我们指出，该方法的一大缺点是计算复杂度较高，这限制了我们在实验中对该方法的使用。

5. 张量分解

张量分解通过近似与时序网络对应的向量空间的基来确定社区结构。该方法将所需的社区数量 η 作为输入，但在真实世界数据中，通常无法预先得知这一数量 [23]。因此，我们在图 (T，η) 中探索参数空间。2我们对张量分解进行随机初始化，并迭代 500 次。我们对前两个因子（张量的 x、y 维度）取平均，再将其与基元素的第三个维度（时间轴）相乘，以获得社区标签。

C. 合成数据生成

我们使用齐次泊松过程模拟了 $N = 78$ 个合成神经元组成的群体的神经元活动，并为每个社区从 4 到 40 之间均匀随机选取放电率。为考虑实验中的随机性，我们研究了 $k = 10$个独立网络。我们使用神经元之间的两两最大互相关来评估神经元放电模式的相似性；也就是说，对该数据集进行动态社区检测的目标，是检测放电模式相似的神经元群体。为了根据实验中的时间序列数据构建时序网络，我们使用了 $\tau = 1000$ 毫秒的窗口大小，并用 $t_{max} = 6$ 创建网络的快照（层）。

为了模拟单调增长社区，我们生成了长度为 $t_{max} \times \tau$ 的主脉冲序列，并随机选取脉冲率，将其抖动 ± 5 毫秒以创建主社区。为了确保结果稳健，主社区的规模从不同分布（均匀分布、高斯分布或指数分布）中随机选取。接下来，每隔 1000 毫秒，我们生成给定规模的独立脉冲序列（规模从与主社区相同的分布中抽取），并使其脉冲活动与主社区同步（同样通过抖动主脉冲序列实现）。这一过程使单例节点在每个时间窗口加入主社区。

对于非单调事件，我们将每个快照中所需的社区数量 N_c 输入时间序列生成流程。对于图中展示的结果，我们为每个快照选取 $N_c = 3$，见图5 和 6。在每一层中，我们创建主脉冲序列，并通过抖动生成相应的社区。随后，我们通过生成独立发放的脉冲序列，将这些社区彼此'拉开'。因此，在每一层中，我们有 N 个神经元，其中一部分被划分到社区中，另一部分为单例节点。如果相邻层中的社区交集超过33%，则为它们分配相同的动态社区标签，以确定其植入社区标签。

生成单调和非单调社区事件的时间序列后，我们将时间序列划分为长度为 $\tau = 1000$的窗口。我们计算每个窗口内的两两最大互相关，构建时序网络的快照表示，以呈现底层预设社区结构。最后，在应用任何动态社区检测方法之前，我们用第一个和最后一个静态快照分别填充每个快照的第一层和最后一层，以避免端点问题。请注意，运行该方法后，填充层中的社区划分会被舍弃。

D. 层间耦合

1. 均匀对角耦合

均匀对角耦合通过在相邻快照中同一节点之间添加无向层间连接来实现，即对于每个节点 $i^t \in V$，其中 $t \in \{1, 2, .., t_{max} - 1\}$，我们赋予一条权重恒定为 ω的层间边（i^t，i^{t+1}）。

2. 对角耦合与局部更新

带局部更新的对角耦合在拓扑上与对角耦合采用相同的技术，但在这种情况下，层间边权重 ω_i^t 可以变化。给定一个恒定的边权重 ω，ω_i^t 当 $\omega.s$ 层 G^t 和 G^{t+1} 之间节点属性的变化小于（或等于）y 个标准差时，等于 ω；如果变化大于 y 个标准差，则等于 [35]，如 $s = \frac{1}{100}$ 所述。我们在实验中使用放电率作为节点属性，并取 $y = 0.5$ 和 $y = 0.5$。

3. 邻域耦合

对于邻域耦合，我们以均匀边权重 ω 连接 i^t 和 i^{t+1}。接下来，我们根据 i^t的层内边强度，确定节点 i^t 在相邻层中的邻域，其中 $t \in \{1, 2, .., t_{max} - 1\}$。我们按照连接强度降序排列 i^t的所有邻居 $\mathbf{N}_i^t$，并仅选取其中排名前 $p\%$ 的邻居 $\{j^{t+1}\}_{j \in \mathsf{N}_i^t}$，作为最大邻居集合；我们将这些邻居与 i^t 耦合，其中 $p = 10$。我们按照与 [39] 中所述的类似流程，归一化层内出链 ω 的权重，并使用詹森–香农散度来确定边权重。具体而言，i^t 和 j^{t+1}之间的层间边权重 $\bar{\omega}_{i^t, j^{t+1}}$为：

$$\bar{\omega}_{i^t, j^{t+1}} = 1 - JSD\left(\frac{\mathsf{G}^t_{(i,j)}}{\sum_{j, j \neq i} \mathsf{G}^t_{(i,j)}}.\omega\right)^2$$

E. 最优区域

我们通过在参数平面内取最大归一化互信息（归一化互信息）的最大值所在的位置，确定方法参数的最优区域，$(T,\ \cdot)$。随后展示该参数组对应的示例划分。尽管相同的最大值可能出现在多个$(T,\ \cdot)$ 参数对中，但由于不同的最大示例划分在结构上通常没有差异，我们仅展示一个示例划分。在图5 和6中，我们通过保持层间边权重 ω不变，并改变层内边阈值 T，来选取非最优区域。

F. 评估划分质量

我们根据真实值（即我们所认为的植入社区标签）比较动态社区检测方法的性能。需要注意的是，由于不同的动态社区检测方法对最优社区的定义各不相同，我们预计不同方法会检测出不同的社区划分。此外，将动态社区检测方法应用于真实世界数据时，并没有已知的真实值可供比较 [49] 因此，我们研究了具有不同植入社区演化的合成数据集。然而，重要的是，在探索不同层间耦合策略时，我们是在单一动态社区检测方法内部比较方法性能；也就是说，我们衡量的是方法性能随层间耦合变化而产生的可量化差异，而不受该方法所用质量函数的影响。我们通过计算真实社区标签 U 与预测社区标签 V 之间的归一化互信息（NMI） [37, 38]:

$$NMI(U,V) = \frac{I(U,V)}{max(H(U),H(V))} \tag{1}$$

其中， $H(\cdot)$ 表示熵， $I(\cdot,\ \cdot)$ 表示互信息；我们选择用标签的最大熵对其进行归一化，因为这种方法更适用于重叠社区 [50]。还需注意，社区标签 U 和 V 针对的是节点—层元素（即每个节点在每一层都有一个社区标签，用以表示其在该层中的社区分配）。此外，我们还在附录 C“划分质量指标”中展示了使用其他指标（调整兰德指数、NVI 和杰卡德指数）衡量划分质量的结果。

致谢

本研究得到美国国家科学基金会资助（SMA-1734795，资助对象为 S. F. Muldoon）。

---

[1] Daniel M. Romero、Brendan Meeder 和 Jon Kleinberg。不同主题下信息扩散机制的差异：Twitter 上的习语、政治标签与复杂传染。载于第 *20* 届万维网国际会议论文集，第 695–704 页，美国纽约州纽约市，2011 年。计算机协会。[2] A. Stopczynski、V. Sekara、P. Sapiezynski、A. Cuttone、M.M. Madsen、J.E. Larsen 等。以高分辨率测量大规模社交网络。《*PLOS ONE*》，9(4):e95978，2014 年。[3] Lijun Sun、Kay W. Axhausen、Der-Horng Lee 和 Xianfeng Huang。理解大都市的日常相遇模式。《美国国家科学院院刊》，110(34):13774–13779，2013 年。[4] E. Valdano、C. Poletto、A. Giovannini、D. Palma、L. Savini 和 V. Colizza。根据以往的时间接触数据预测流行病风险。*PLoS Computational Biology*，11(3):e1004152，2015 年。[5] D. Kondor、M. Posfai、I. Csabai 和 G. Vattay。富者愈富吗？比特币交易网络的实证分析。《*PLOS ONE*》，9(2):e86197，2014 年。[6] I. Scholtes、N. Wider 和 R. et al. Pfitzner。非马尔可夫时序网络中由因果关系驱动的扩散减速与加速。*Nature Communications*，5:5024，2014 年。[7] M. E. J. Newman。科学合作网络的结构。《美国国家科学院院刊》，98 409，2001 2 404( ): – 。[8] M. Rosvall、A. Esquivel 和 A. et al. Lancichinetti。网络流中的记忆及其对传播动力学和社区检测的影响。*Nature Communications*，5:4630，2014 年。[9] Taylor I.、Linding R. 和 Warde-Farley D. et al.。蛋白质相互作用网络中的动态模块度可预测乳腺癌结局。*Nature Biotechnology*，27:199–204，2009 年。[10] Rasmussen C、Dupont YL、Mosbacher JB、Trøjelsgaard K 和 Olesen JM。时间分辨率对生态网络结构的显著影响。《*PLOS ONE*》，8(12):e81694，2013 年。

[11] Hae-Jeong Park 和 Karl Friston。结构与功能脑网络：从连接到认知。《科学》，342 1238411，2013 6158( ): 。[12] D.S. Bassett、N.F. Wymbs、M.P. Rombach、M.A. Porter 和 P.J. et al. Mucha。基于任务的核心—边缘组织与人脑动力学。PLoS Computational Biology，9(9):e1003171.，2013。[13] Naoki Masuda 和 Renaud Lambiotte。《时序网络指南》。新加坡：世界科学出版社，2016。[14] P. Holme。现代时间网络理论：一篇专题综述。《欧洲物理学杂志B辑》，88:234，2015。[15] R´emy Cazabet 和 Fr´ed´eric Amblard。《动态社区检测》。施普林格纽约，纽约州纽约市，2014。[16] Giulio Rossetti 和 R´emy Cazabet。动态网络中的社区发现：综述。《ACM计算调查》，51(2)，2018年2月。[17] P. J. Mucha、T. Richardson、K. Macon、M. A. Porter 和 J.-P. Onnela。时变、多尺度和多重网络中的社区结构。《科学》，328(5980):876–8，2010。[18] Martin Rosvall 和 Carl T. Bergstrom。复杂网络中的随机游走图揭示社区结构。《美国国家科学院院刊》，105(4):1118–1123，2008。[19] Martin Rosvall、Daniel Edler、Anton Eriksson。Infomap Python API。https://mapequation.github.io/infomap/python/。[20] Tiago P. Peixoto。推断分层、边赋值及时变网络的中尺度结构。《物理评论E》，92:042807，2015年10月。[21] Emmanuel Abbe。社区检测与随机块模型：近期进展。《机器学习研究杂志》，18(177):1–86，2018。[22] LE. Martinet、M.A. Kramer 和 W. et al. Viles。鲁棒动态社区检测及其在人脑功能网络中的应用。Nature Communications，11:2785，2020。[23] L. Gauvin、A. Panisson 和 Cattuto C.。检测时序网络的社区结构和活动模式：一种非负张量分解方法。《PLOS ONE》，9(1)，2014。[24] N. Otter、M.A. Porter 和 U. et al. Tillmann。持久同调计算路线图。《EPJ数据科学》，6(17)，2017。[25] Ann E. Sizemore、Jennifer E. Phillips-Cremins、Robert Ghrist 和 Danielle S. Bassett。整体的重要性：面向网络神经科学家的拓扑数据分析。《网络神经科学》，3(3):656–673，2019年7月。[26] Chad Giusti、Robert Ghrist 和 Danielle S Bassett。两者成伴，三者（或更多）成单纯形。《计算神经科学杂志》，41(1):1–14，2016。[27] A.E. Sizemore、C. Giusti 和 A. et al. Kahn。人类连接组中的团与空腔。《计算神经科学杂志》，44:115–145，2018。[28] Clara Granell、Richard K. Darst、Alex Arenas、Santo Fortunato 和 Sergio G´omez。用于评估演化网络社区结构的基准模型。《物理评论E》，92:012805，2015年7月。[29] Cazabet Remy、Boudebza Souˆaad 和 Rossetti Giulio。评估渐进演化图的社区检测算法。《复杂网络杂志》，8，2020年12月。[30] Rossetti Giulio。Rdyn：处理社区动力学的图基准。《复杂网络杂志》，5:893–912，2017年12月。[31] Marya Bazzi、Lucas G. S. Jeub、Alex Arenas、Sam D. Howison 和 Mason A. Porter。用于构建多层网络中尺度结构生成模型的框架。《物理评论研究》，2:023100，2020年4月。[32] Mel MacMahon 和 Diego Garlaschelli。相关矩阵的社区检测。《物理评论X》，5:021006，2015年4月。[33] Marya Bazzi、Mason A. Porter、Stacy Williams、Mark McDonald、Daniel J. Fenn 和 Sam D. Howison。时间多层网络中的社区检测及其在相关网络中的应用。《多尺度建模与仿真》，14(1):1–41，2016。[34] J. O. Garcia、A. Ashourvan、S. F. Muldoon、J. M. Vettel 和 D. S. Bassett。社区检测技术在脑图中的应用：算法考量及其对神经功能的启示。《IEEE会刊》。电气与电子工程师学会，106(5):846–867，2018。[35] Sarah F Muldoon Michael Vaiana、Ethan M Goldberg。通过动态社区检测优化功能性时序网络中的状态变化检测。《复杂网络杂志》，7:529–553，2019年8月。[36] Manlio De Domenico、Andrea Lancichinetti、Alex Arenas 和 Martin Rosvall。识别多层网络中的模块化流，揭示互联系统中高度重叠的组织结构。《物理评论X》，5:011027，2015年3月。[37] Leon Danon、Albert D´ıaz-Guilera、Jordi Duch 和 Alex Arenas。比较社区结构识别方法。《统计力学杂志：理论与实验》，2005(09):P09008–P09008，2005年9月。[38] Marina Meil˘a。比较聚类——一种基于信息的距离。《多元分析杂志》，98(5):873–895，2007。[39] Ulf Aslak、Martin Rosvall 和 Sune Lehmann。时序网络中受约束的信息流揭示间歇性社区。《物理评论E》，97:062312，2018年6月。[40] William H. Weir、Benjamin Walker、Lenka Zdeborov´a 和 Peter J. Mucha。用于评估社区结构可检测性的多层模块度置信传播。《SIAM数据科学数学杂志》，2(3):872–900，2020。[41] Tiago P. Peixoto。揭示网络划分之间的共识与分歧。《物理评论X》，11:021003，2021。[42] Bengier ¨Ulgen Kılıc。GitHub，时序网络分析。https://github.com/ulgenklc/temporal_network_ analysis。[43] Bengier ¨Ulgen Kılıc。Read the Docs，时序网络分析。https://temporal-network-analysis.readthedocs. io/en/latest/index.html。[44] J¨org Reichardt 和 Stefan Bornholdt。社区检测的统计力学。《物理评论E》，74:016110，2006年7月。[45] V.A. Traag、L. Waltman 和 N.J. van Eck。从 Louvain 到 Leiden：确保社区具有良好连通性。《科学报告》，9 5233，2019 : 。[46] V. D. Blondel、J.-L. Guillaume、R. Lambiotte 和 E. Lefebvre。大型网络中的社区快速展开。《统计力学杂志：理论与实验》，10008(10):6，2008。[47] Tiago P. Peixoto。graph-tool API。https://graph-tool.skewed.de。

[48] G. Palla、I. Der´enyi 和 I. 等人 Farkas。揭示自然与社会中复杂网络的重叠社区结构。《自然》，435:814–818，2005。[49] Leto Peel、Daniel B. Larremore 和 Aaron Clauset。关于网络中元数据与社区检测的真实情况。《科学进展》，3(5):e1602548，2017。[50] Nguyen Xuan Vinh、Julien Epps 和 James Bailey。用于聚类比较的信息论度量：变体、性质、归一化与偶然性校正。《机器学习研究杂志》，11(95):2837–2854，2010。

附录 A：骨架耦合算法

本节给出骨架耦合算法的伪代码。请注意，所提供的算法并非优化版本，但其时间复杂度至多为 O($N^2.t_{max}$)。

设 $P_t = \{C_1^t, C_2^t, ..., C_p^t\}$ 和 $P_{t+1} = \{C_1^{t+1}, C_2^{t+1}, ..., C_r^{t+1}\}$ 分别为通过节点对齐时序网络 $T = \{G^1, G^2, ..., G^{t_{max}}\}$的静态社区检测方法获得的快照 G^t 和 G^{t+1}的划分。从节点 i^t 到快照 G^{t+1} 中各节点的层间连接，可通过“算法 1”的输出获得；字典 $SC[i,t]$。

算法 1骨架耦合

对于 $t = 1$ **到** $t_{max} - 1$ **执行**
  对于 $\alpha = 1$ **到** p **执行**
    对 i 中的 C_α^t **do**
      查找 β 使得 $i \in C_\beta^{t+1}$
      如果 $|C_\alpha^t| > 1$ **则**
        如果 $|C_\beta^{t+1}| > 1$**则**
          $SC[i,t] \longleftarrow [C_\beta^{t+1}]$ {case iv}
        else
          $SC[i,t] \longleftarrow []$ {case iii}
        结束如果
      else
        如果 $|C_\beta^{t+1}| > 1$**则**
          $SC[i,t] \longleftarrow []$ {case ii}
        else
          $SC[i,t] \longleftarrow [i^{t+1}]$ {case i}
        结束条件判断
      结束条件判断
    结束循环
  结束循环
结束循环

附录 B：更多社区演化情景

在正文中，我们研究了两种社区演化情景（即单调事件和非单调事件），以比较图中所提出的骨架耦合算法与 MMM 和 Infomap 方法相结合时的性能。5和6中的结果。本节中，我们在植入了动态社区结构的时间序列上补充了类似的实验，该结构代表单调事件和非单调事件。

其中第一个实验仍然是单调增长情景：如“神经元数据模拟”一节和“合成数据生成”方法部分所述，初始主社区在每个时间窗口都会吸纳一组单例社区。不同的是，这次不断增长的主社区在时间序列结束时并未吸纳所有神经元。真实标签见图 B.1 和 B.2 的 A 面板。B.1 和B.2。

第二个实验与我们在“合成数据生成”方法部分中描述的非单调事件完全相同。不过，这次我们在图 B.1 和 B.2 的 B 面板中选择 $N_c = 2$ 。这样，在两个情景中保持层数和神经元总数不变的同时，我们获得了不同的社区规模与网络规模之比。B.1 和B.2。这样做可以在保持两个情景的层数和神经元总数相同的情况下，得到不同的社区规模与网络规模之比。

附录 C：划分质量指标

在正文中，我们使用归一化互信息（归一化互信息）作为主要指标，研究划分与真实值之间的关系。我们通过研究图中考察的参数空间，检验骨架耦合结果的稳健性。5和6中的参数空间，并在图C.1 和C.2中分别使用调整兰德指数（调整兰德指数）、标准化变异信息（标准化变异信息）和杰卡德指数进行分析。结果表明，就这些划分质量指标而言，最优区域和非最优区域与使用归一化互信息得到的区域一致。因此，正文中呈现的结果仍然是稳健的。

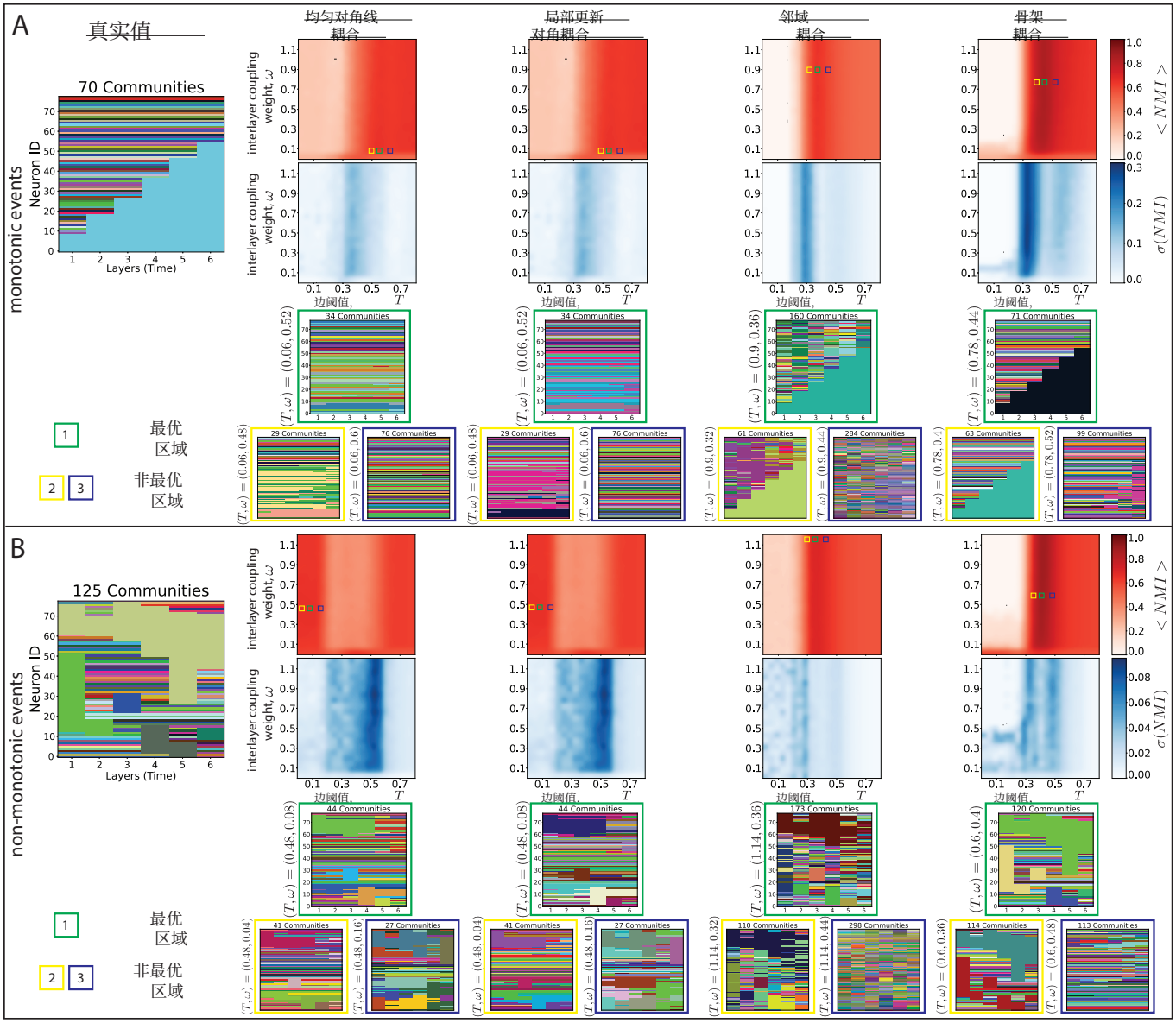

图 B.1. **在更多社区演化情景下，不同 MMM 层间耦合策略的比较**。对于经历**A.** 扩展型社区事件的 $N = 78$个神经元（共 70 个社区标签）以及**B.** 经历瞬态社区事件的神经元（共 125 个社区标签），比较不同的层间耦合策略。社区标签以颜色表示。参数景观展示了不同随机运行中，方法性能的平均归一化互信息（$< NMI >$）及标准差（$\sigma(NMI)$），在（T，ω）参数空间中的表现。A 中扩展型事件的零模型采用配置模型，分辨率参数为 0。98；B 中瞬态事件的分辨率参数为 1。3。在每个参数空间下，我们展示 MMM 方法在最优区域和非最优区域范围内找到的示例共识划分；这些区域分别以绿色、黄色和蓝色矩形标出，并按归一化互信息由高到低排列。

附录 D：随机运行之间的最大重叠距离

正如正文中所述，我们根据不同随机运行中的共识算法 [41]确定最优区域和非最优区域。在图D.1 和D.2中，我们研究网络每次随机生成的最优划分之间的最大重叠距离（最大重叠距离） [41]。最大重叠距离越高，两个划分就越相似。每个网络对应不同的时间序列，其脉冲率各不相同，但植入的动态社区结构相同。

在图 D.1中，我们展示了通过 MMM 方法获得的、具有相同社区结构的独立生成时间序列各自最优划分之间的 MOD 距离。从上到下的各个面板中，我们分别关注不同的社区

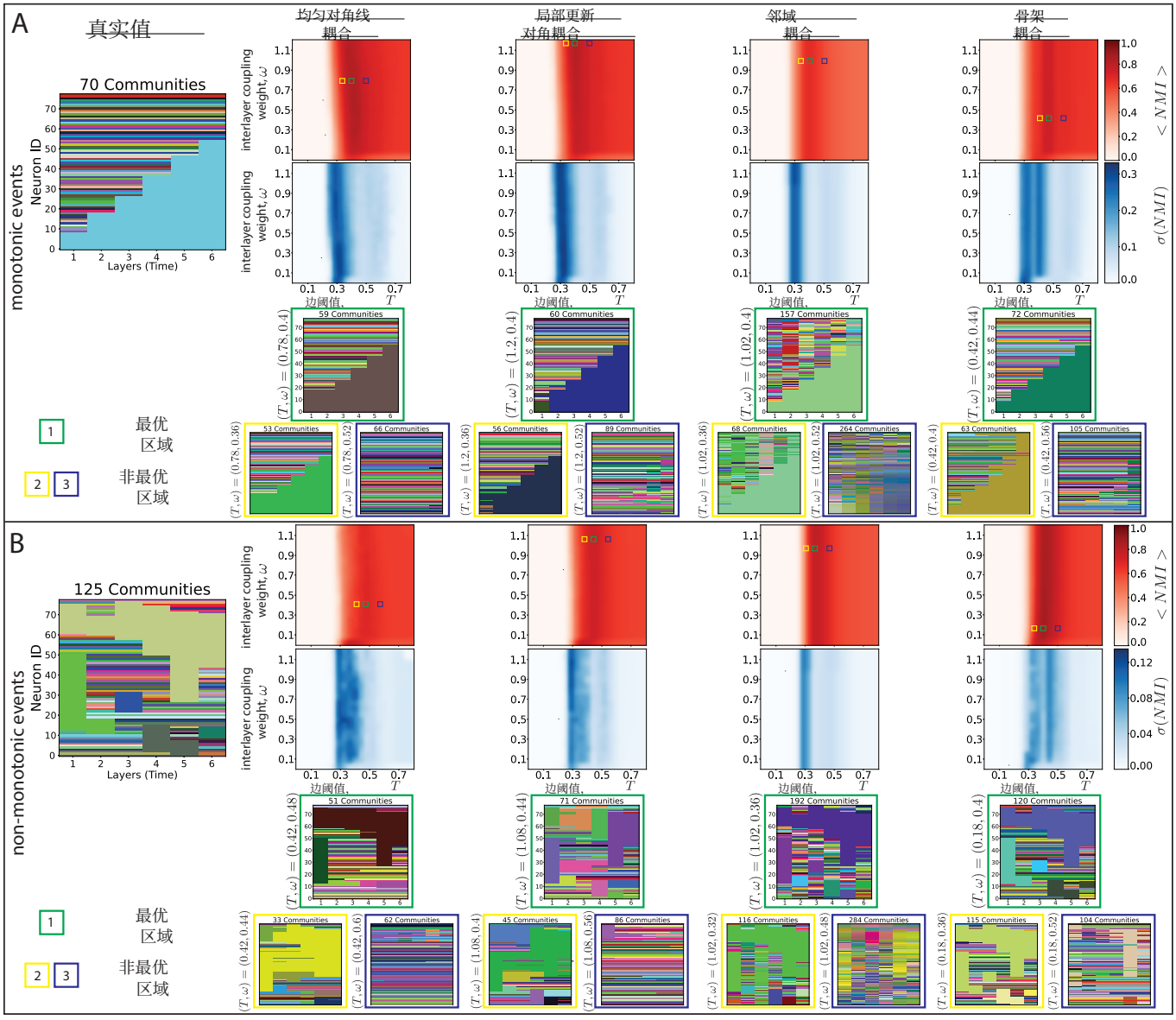

图 B.2。**针对其他社区演化情景，比较 Infomap 的不同层间耦合策略**。比较对经历**A.** 扩展型社区事件且共有 70 个社区标签的 $N = 78$ 神经元，以及经历**B.** 瞬态社区事件且共有 125 个社区标签的神经元所采用的不同层间耦合启发式方法。社区标签以颜色表示。参数景观展示了在（T，ω）参数空间中，不同随机运行下通过平均 NMI（$< NMI >$）和标准差（$\sigma(NMI)$）衡量的方法性能。两个面板中的多层松弛率均设为 0。2。在每个参数空间下，我们按照 NMI 从高到低的顺序，分别展示了 Infomap 方法在绿色、黄色和蓝色矩形标出的最优区域和非最优区域范围内得到的示例共识划分。

演化情景，分别见图5A)、图5B)、图B.1A) 和图B.1B)。可以观察到，在每种情景下，对角耦合方法的 MOD 距离都相当低，邻域耦合的平均 MOD 距离适中，而骨架耦合的则较高。这些发现表明，MMM 与骨架耦合相结合，即使基准网络具有随机性，也能获得稳健的结果。

接下来，在图 D.2中，我们展示完全相同的结果，但这次针对的是 Infomap。类似地，从上到下面板，我们分别关注图6A）、图6B）、图B.2A）和图B.2B）中研究的不同社区演化情景。尽管在此情况下很难比较不同启发式方法，并推断最优划分之间产生的 MOD距离，但我们注意到，Infomap 与骨架耦合相结合时，其 MOD距离并不低，这仍表明结果具有相对较强的稳健性。

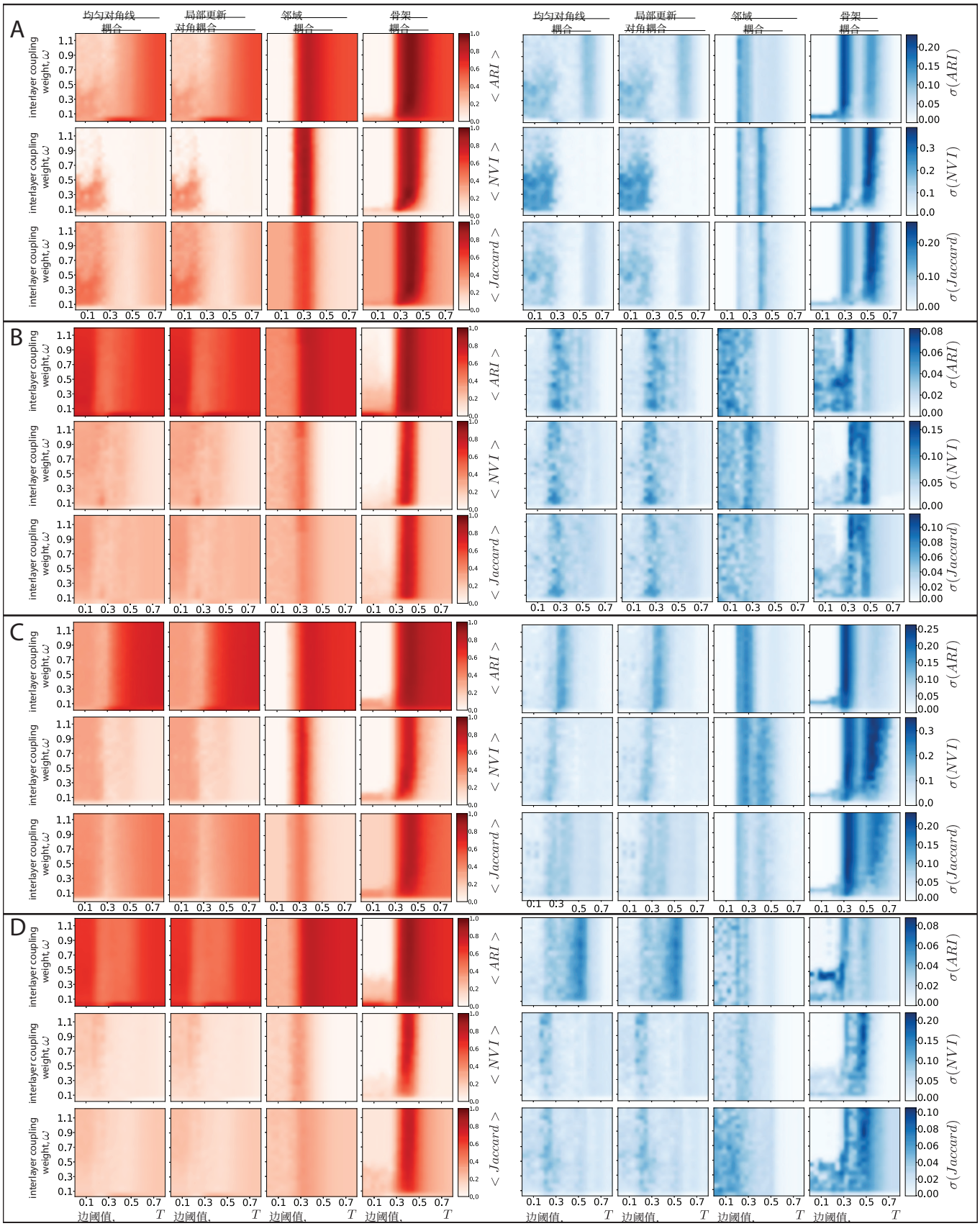

图 C.1。 **使用 MMM 比较不同信息度量和层间耦合策略下的参数景观**。 我们通过计算 ARI、归一化变异信息和杰卡德指数，将所得划分与真实值进行比较，以此研究这些划分。 **A.** 图5中的单调事件。 **B.** 图5中的非单调事件。 **C.** 图B.1中的单调事件。 **D.** 图B.1中的非单调事件。

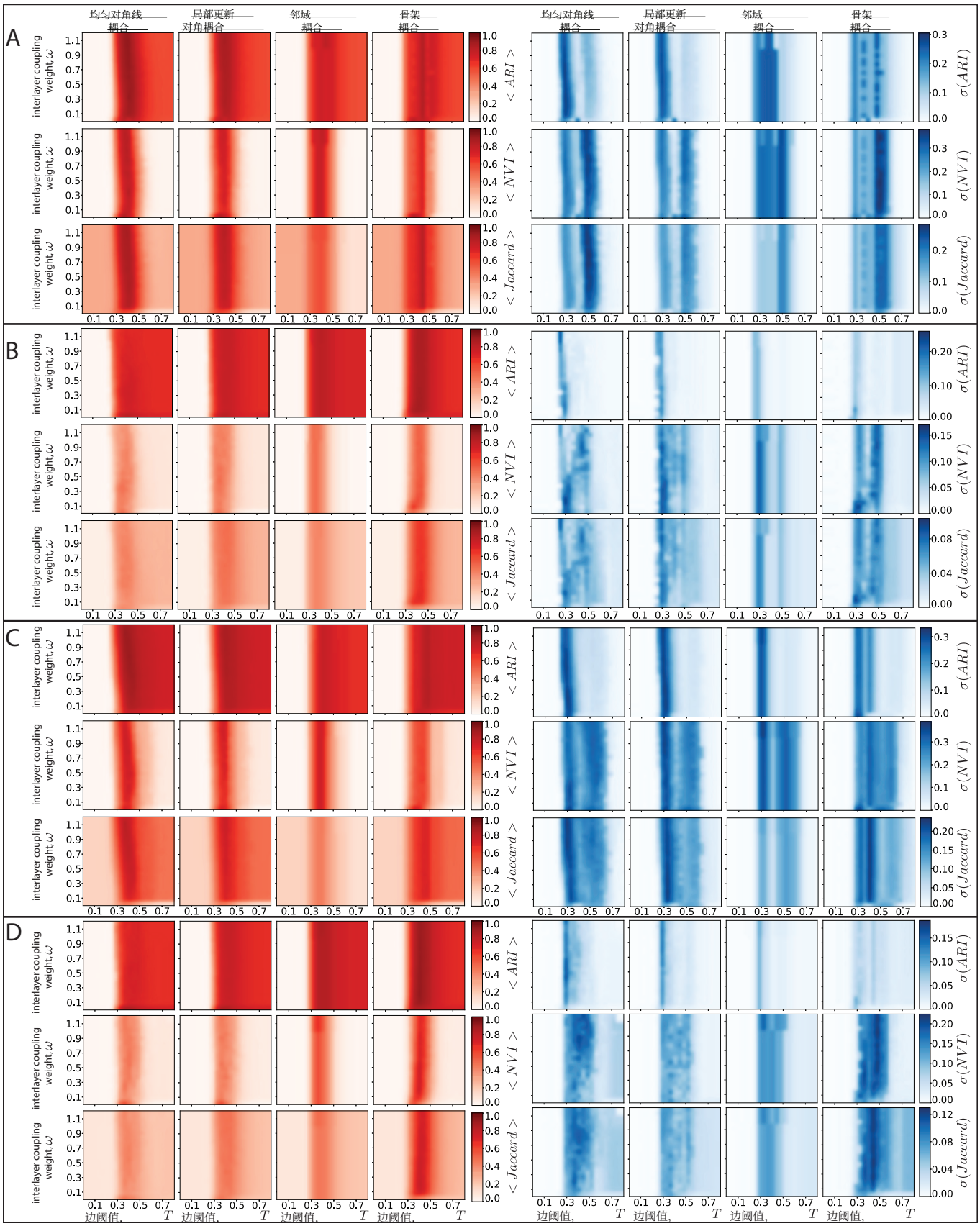

图 C.2。 **使用 Infomap 比较不同信息度量和层间耦合策略对应的参数景观。** 我们通过调整兰德指数、归一化变异信息和杰卡德指数，将得到的划分与真实值进行比较。 **A.** 图 6中的单调事件。 **B.** 图 6中的非单调事件。 **C.** 图 B.2中的单调事件。 **D.** 图 B.2中的非单调事件。

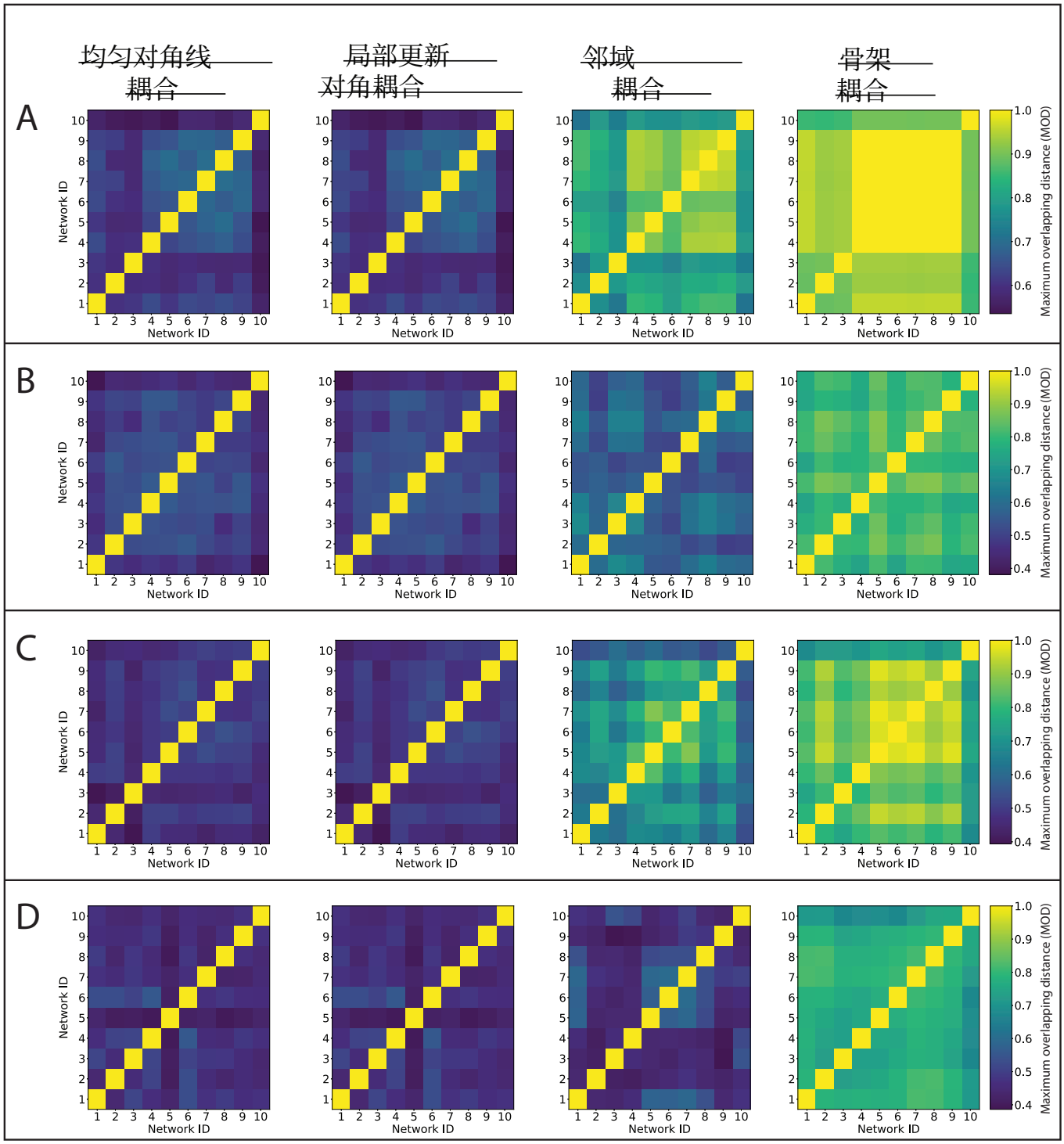

图 D.1. **MMM 在每次随机运行中的最优划分之间的最大重叠距离**。 所有随机运行中最优划分之间的最大重叠距离（MOD）：**A.** 图中的单调事件5。 **B.** 图中的非单调事件5。 **C.** 图中的单调事件B.1。 **D.** 图中的非单调事件B.1。

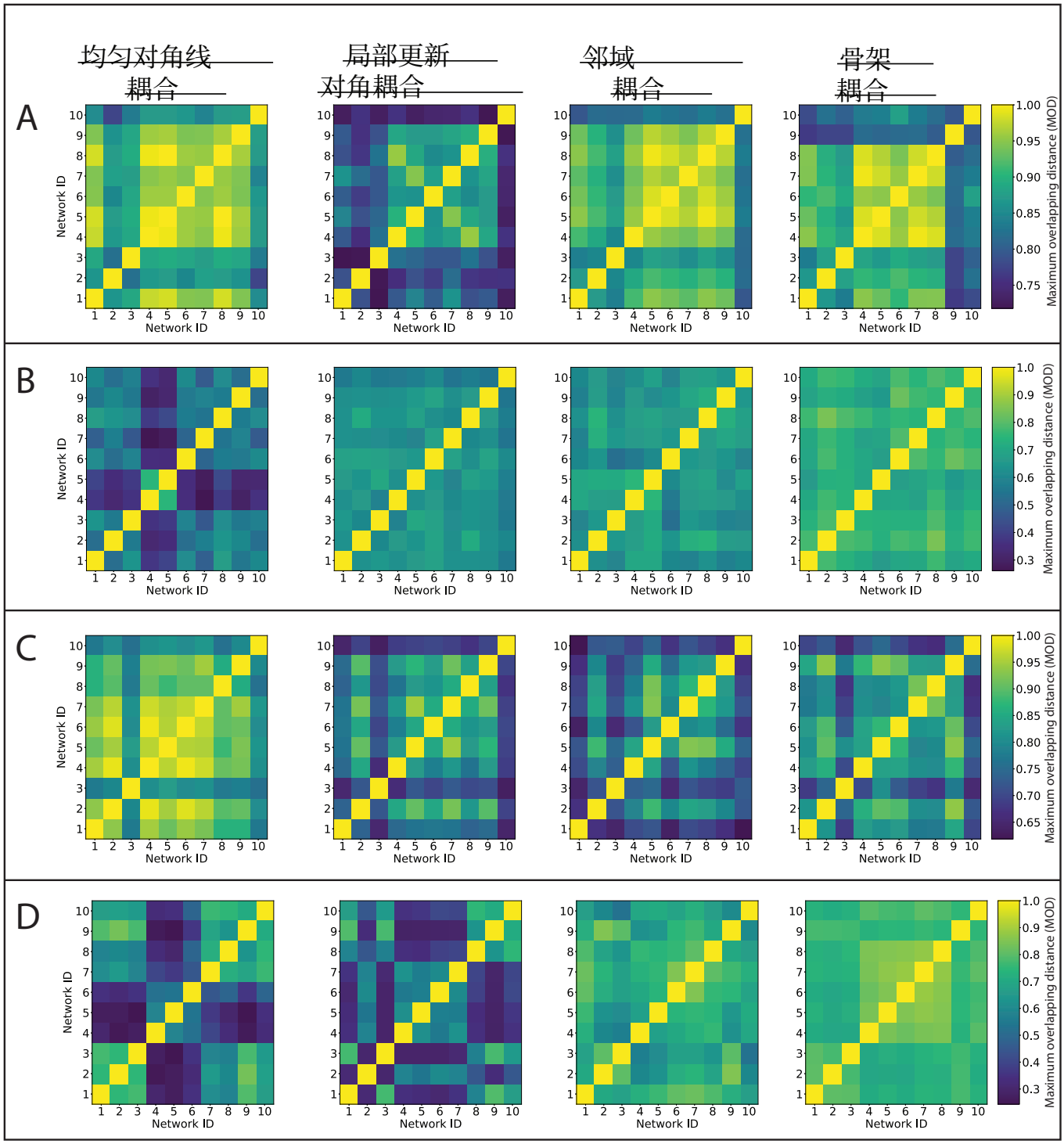

图 D.2. **每次随机运行中 Infomap 最优划分之间的最大重叠距离。** 所有随机运行中最优划分之间的最大重叠距离（MOD），对应于 **A.** 图中的单调事件6。 **B.** 图中的非单调事件6。 **C.** 图中的单调事件B.2。 **D.** 图中的非单调事件B.2。