

# Arquitectura Möbius-RAS-Sinhome (GDL & Física)

## 0) Resumen Ejecutivo

- **Topología avanzada:** La tira de Möbius es un *manifold* 2D no orientable con un solo borde <sup>1</sup>. Parametrizar un grafo sobre esta superficie permite explotar conexiones geométricas inusuales (folding) que crean atajos *extrínsecos* reduciendo distancias de camino medio, similar al efecto de las aristas de largo alcance en redes *small-world* <sup>2</sup>.
- **Dinámica de grafos:** Modelar la rotación y fuerzas en el grafo equivale a procesos de difusión de señal sobre el *graph* (p.ej. usando ecuaciones de difusión basadas en el laplaciano) <sup>3</sup>. Esto suaviza y expulsa el “ruido” (nodos de alta entropía) hacia la periferia, análogo a un flujo de difusividad en el grafo.
- **Agente de control RAS:** Un mecanismo de *gating* (RAS/Sinhome) activaría selectivamente atajos según métricas de coherencia local (p.ej. atención GNN o similitud de features <sup>4</sup>) o minimización de energía libre (principio de Friston). Esto introduce conectividad dinámica, emulando esquemas de redes “de apuntadores” (pointer networks) <sup>5</sup>.
- **Análogos físicos:** Conceptos de gravedad y relatividad sirven de inspiración: más densidad de información actúa como pozo gravitatorio que exige más “pasos de cómputo” (mediante ACT) <sup>6</sup>, mientras que rotación implica difusividad. Estas analogías pueden traducirse a algoritmos de *Adaptive Computation Time* y optimización riemanniana adaptativa.
- **Viabilidad GDL:** El marco propuesto fusiona teoría establecida (GNNs en variedades, small-world graphs, atención en grafos) con hipótesis simulables (atajos geométricos, dinámicas físicas en grafos) y especulaciones inspiradas en física. Aunque experimental, aprovecha principios de GDL modernos <sup>4</sup> <sup>3</sup>. Su viabilidad depende de implementaciones cuidadosas en PyTorch Geometric y validaciones empíricas.

## 1) Estado Epistemológico

- **(A) Manifold de Möbius:** La tira de Möbius es una superficie 2D no orientable (un solo lado y un borde) <sup>1</sup>. Se puede parametrizar con  $x(u,v), y(u,v), z(u,v)$ ,  $u \in [0, 2\pi]$ ,  $v \in [-w, w]$ , como en textos clásicos de geometría diferencial.
- **(B) Atajos por folding:** Plegar la banda en 3D crea atajos extrínsecos entre nodos alejados en la coordenada  $u$  (análogo a “rewiring” en small-world) <sup>2</sup>. Esto es una hipótesis comprobable: los atajos reducen la distancia de camino medio en el grafo.
- **(A) Redes Small-World:** En redes “pequeño mundo” los atajos aleatorios reducen drásticamente las distancias de red <sup>2</sup>. Aquí los atajos son deterministas según la geometría, pero cumplen papel análogo (bajo conteo de distancias).
- **(B) Métrica RAS – Atención:** Proponemos usar métricas de similitud entre features de nodos para activar atajos (p.ej. producto punto con atención tipo GAT) <sup>4</sup>. Alternativamente, un principio de “energía libre” (energetic models) guiaría la coherencia de conexiones.
- **(A) Difusión en grafos:** La ecuación de difusión laplaciana en grafos suaviza señales de nodo <sup>3</sup>. Esto es equivalente a simular difusividad física: nodos similares intercambian valor, bajando entropía global.

- **(C) Centrifugado de ruido:** Intuimos que la rotación ejerce fuerza centrífuga que expulsa nodos ruidosos hacia los bordes, similar a un flujo de Ricci gráfico. Especulación física que sugiere usar flujos de curvatura del grafo (Ricci-Ollivier) para redistribuir inhomogeneidades.
- **(A) Adaptive Computation Time:** ACT permite a una RNN/GNN decidir cuántos pasos de inferencia usar por entrada, dedicando más pasos a casos complejos <sup>6</sup> <sup>7</sup>. En analogía, la “gravidad” (alta densidad informativa) ralentizaría el tiempo de inferencia (mayor cómputo).
- **(B) Optimización Riemanniana:** La idea de optimizar sobre variedades (Riemannian SGD, etc.) es establecida en ML <sup>8</sup>. Aquí la métrica “gravitacional” podría modular la tasa de aprendizaje local (menor LR en zonas densas) como forma de dilatación temporal.
- **(A) Pointer Graph Networks:** Los PGN permiten que cada nodo apunte dinámicamente a otro para paso de mensajes <sup>5</sup>. Esta es la base para punteros cíclicos en  $\mathcal{G}$ : nodos o marcadores que avanzan a lo largo de  $\mathcal{G}$  conectando dinámicamente componentes.
- **(A) Möbius vs. Borromean:** Una tira de Möbius es una superficie (2D) con un lazo de borde <sup>1</sup>, mientras que los anillos Borromeos son tres lazos enlazados topológicamente <sup>9</sup>. Son estructuras distintas, aunque en principio un grafo con atajos podría imitar funcionalmente la estabilidad en tres ciclos enlazados bajo control del RAS.

## 2) Registro de Evidencia

Claim_ID	Claim	Etiqueta	Idea Clave ML/Math	Riesgo	Cómo Verificar (Simulación)
C1	La tira de Möbius es un <i>manifold</i> no orientable con un único borde <sup>1</sup> .	A	Variedades no euclídeas	Fallos en los bordes (continuidad)	Implementar grafo con $u \in [0, 2\pi]$ , $v \in [-w, w]$ y verificar condiciones de borde.
C2	Plegar la banda genera <i>atajos</i> extrínsecos que reducen la distancia de camino medio (como un rewire geométrico) <sup>2</sup> .	B	Redes small-world	Sobreconexión, pérdida de localización	Simular gráfica 2D vs Möbius plegado, comparar longitud promedio de caminos.
C3	A diferencia de Small-World aleatorio, los atajos aquí son <i>deterministas</i> por geometría <sup>2</sup> .	A	Modelo Watts-Strogatz	Posible baja aleatoriedad inductiva	Comparar estadística de clustering y path length con modelo WS.

Claim_ID	Claim	Etiqueta	Idea Clave ML/Math	Riesgo	Cómo Verificar (Simulación)
<b>C4</b>	Usar atención basada en similitud (dot-product) entre nodos para activar atajos (inspirado en GAT) <sup>4</sup> .	A	GNN Atención	Vínculos irrelevantes (ruido en features)	Entrenar GAT en tareas de clasificación con/sin gating de atajos.
<b>C5</b>	Proponer métrica de “energía libre” para gating (minimizar incoherencia local) (Friston).	C	Principio de energía libre	Difícil de definir objetivamente	Experimentar con métricas como entropía local o pérdidas variacionales para gating.
<b>C6</b>	Rotación $\approx$ proceso de difusión en el grafo, suavizando señal y llevando ruido a bordes.	B	Difusión Laplaciana en grafos <sup>3</sup>	Sobre-suavizado (over-smoothing)	Simular grafo rotacional, medir varianza de features vs tiempo; comparar con diffusion PDE.
<b>C7</b>	La difusión en grafos es matemática y práctica establecida (p.ej. GRAND, Graph Laplaciano) <sup>3</sup> .	A	Ecuación de difusión PDE	Computación costosa (PDE implícita)	Implementar GRAND u otras GNN basadas en difusión y medir mejora de ruido.
<b>C8</b>	Fuerza centrífuga expulsa nodos de alta entropía: analogía con <i>Ricci flow</i> para regularizar curvatura de grafo.	C	Flujos de Ricci en ML <sup>10</sup>	Analogía débil; interpretación subjetiva	Calcular curvatura Ollivier en el grafo y medir cambios tras difundir características.

Claim_ID	Claim	Etiqueta	Idea Clave ML/Math	Riesgo	Cómo Verificar (Simulación)
<b>C9</b>	La “gravedad” de alta densidad informativa exige más pasos de inferencia (Adaptive Computation Time) <sup>6</sup> <sup>7</sup> .	A	Adaptive Computation Time (ACT)	Incremento de cómputo (latencia)	Implementar ACT en GNN (p.ej. variable de decisión de pasos) y evaluar calidad vs costo.
<b>C10</b>	Analogía: Optimización en variedad (Riemannian) con tasa de aprendizaje adaptativa según curvatura/ información.	C	Optimización Riemanniana <sup>8</sup>	Definición de “gravedad” imprecisa	Comparar SGD vs Riemannian SGD en entrenamiento GNN con base topológica.
<b>C11</b>	<i>Pointer Graph Networks</i> : cada nodo puede apuntar dinámicamente a otro, formando enlaces adaptativos <sup>5</sup> .	A	Redes de punteros (PGN)	Complejidad de diseño	Emplear PGN en un grafo trivial; observar si aprende estructura global.
<b>C12</b>	Marcadores de ruleta recorren \$u\$ cíclicamente, guiando E/S dinámicamente (inspirado en Pointer Networks).	B	Pointer Networks	Patrón de movimiento difícil de sincronizar	Simular marcadores recorrientes en distintas velocidades y medir flujo de información.
<b>C13</b>	Möbius $\neq$ Borromeos: Möbius es superficie de 1 lazo <sup>1</sup> ; Borromeos son 3 lazos enlazados <sup>9</sup> .	A	Topología de superficies vs enlaces	Malinterpretación topológica	Validar Euler característico; asegurarse de la topología diferente.

Claim_ID	Claim	Etiqueta	Idea Clave ML/Math	Riesgo	Cómo Verificar (Simulación)
C14	Hipótesis: Un solo manifold con atajos (Möbius) podría emular funcionalidad estable de 3 anillos Borromeos bajo control RAS.	C	Analogía topológica-funcional	Muy especulativo	Diseñar escenario de 3 "regiones" en Möbius y probar si RAS las conecta como anillos Borromeos.

### 3) Diseño de la Propuesta

- **Ecuación del Manifold:** Parametrizamos la tira de Möbius como  $x(u,v) = \text{bigl}(1 + \frac{v}{2}\cos\frac{u}{2}\bigr)\cos u, \text{quad } y(u,v) = \text{bigl}(1 + \frac{v}{2}\cos\frac{u}{2}\bigr)\sin u, \text{quad } z(u,v) = \frac{v}{2}\sin\frac{u}{2}$ , donde  $u \in [0, 2\pi]$ ,  $v \in [-1, 1]$  (ancho de la banda). Cada nodo  $i$  tiene coordenadas  $(u_i, v_i)$ , y el grafo base conecta vecinos en  $(u,v)$  (p.ej. reticulado rectangular). La no orientabilidad se maneja enlazando  $u=0$  con  $u=2\pi$  con inversión de signo en  $v$ .
- **Mecánica del Sinthome (RAS):** Proponemos un algoritmo de *gating* para aristas atajo: para cada par de nodos potencialmente cercanos en 3D tras el folding, calculamos una **función de atención** basada en sus embeddings de nodo. Por ejemplo,  $a_{ij} = \text{softmax}(W_K h_i \cdot W_Q h_j / \sqrt{d})$  <sup>4</sup>. Si  $a_{ij}$  supera un umbral adaptativo (o minimiza energía libre combinando similitud semántica y coste de edge), se activa la arista dinámica  $i \rightarrow j$ . Este RAS permite que la conectividad evolucione durante el entrenamiento, preservando coherencia local (similares conectados) y reduciendo la complejidad global.
- **Dinámica Física:**
  - **Rotación = Difusión:** Al simular la banda girando, los valores en nodos se propagan por efecto centrífugo. Algoritmicamente, esto equivale a una iteración de **difusión en el grafo**:  $X(t+1) = X(t) + \alpha L X(t)$ , donde  $L$  es el Laplaciano normalizado. Este paso suaviza las características (disminuyendo ruido) como en GRAND <sup>3</sup>.
  - **Gravedad = LR Adaptativo:** El "centro" de la banda (p.ej.  $u=\pi$ ) puede asignar mayor densidad informativa. Implementamos un **Adaptive Computation Time (ACT)**: cada nodo/modelo tendrá un contador de pasos de mensaje que depende de su entropía local o centralidad. Alternativamente, usamos **optimización riemanniana** con tasa de aprendizaje  $\eta_i \propto 1/(1 + \text{densidad}_i)$ , de modo que regiones densas ("gravitacionales") se entrenan con LR menor (dilatan el tiempo de aprendizaje). Esto replica la idea de dilatación temporal en regiones complejas <sup>6</sup> <sup>7</sup>.

### 4) Plan de Simulación

Proponemos tres configuraciones experimentales en PyTorch Geometric (u otra librería GNN):

1. **Baseline:** Grafo plano (cilíndrico sin twist) con topología local estándar (vecinos en cuadrícula) y sin dinámica física. Entrenar un GNN convencional (p.ej. GCN o GAT).

2. **Möbius-Folded (estático):** Grafo en tira Möbius con atajos fijos: añadimos aristas entre nodos cuya distancia euclídea 3D sea menor que un umbral. Entrenar el mismo GNN pero con estas aristas extra.
3. **Möbius-Rotational (dinámico):** Igual que Möbius-Folded, pero además simulamos dinámicamente rotación: en cada época actualizamos las posiciones 3D (rotándolas) y volvemos a calcular aristas según nuevo plegado. Paralelamente, aplicamos el gating RAS para activar/desactivar atajos en cada step. Incluimos pasos de difusividad y ACT en el forward.

Compararemos estas configuraciones en tareas de regresión/clasificación espacial (p.ej. propagación de señales en una malla, identificación de patrones en la superficie). Métricas: precisión de predicción, entropía de nodos, longitud media de caminos, estabilidad de gradiente. Se usarán redes con/ sin capas de atención (para gating).

## 5) Próximos Pasos y Riesgos

- **Implementación detallada:** Traducir las ideas físicas a código requiere desarrollar módulos personalizados en PyTorch Geometric: crear el manifold Möbius, funciones de gating basadas en atención y energía libre, capas de difusión (p.ej. `GraphLaplaceConv` o integración ODE), y mecanismos ACT en el grafo.
- **Complejidad computacional:** Las aristas dinámicas y el ACT aumentan el costo. Riesgo de bucle de feedback (ciclos de atajos). Se mitigará limitando densidad de atajos y usando entrenamiento por fases (primero estático luego dinámico).
- **Hiperparámetros por robustez:** Ajustar umbrales de gating, coeficientes de difusión y ritmos de aprendizaje adaptativos puede ser complejo. El riesgo es que el modelo aprenda triviales (p.ej. apagar todos los atajos). Se verificará cada componente aislado antes de integración total.
- **Validación de topología:** Debe verificarse que el grafo preserve la conectividad esperada de la Möbius (no se “rompa” el borde) y que los atajos realmente reproduzcan la analogía Borromean sugerida. Se usarán tests topológicos (características de Euler, conectividad de subgrafos) durante la experimentación.
- **Entorno:** Se sugiere usar PyTorch Geometric o DGL para flexibilidad en grafos dinámicos. Se desarrollará en Python 3 con GPU, midiendo rendimiento y escalabilidad (número de nodos vs tiempo).
- **Futuras iteraciones:** Si los resultados iniciales son prometedores, se explorarán variantes: bandas con más twists, otros *manifolds* (Klein bottle) o distintos mecanismos de Sinhome inspirados en sistemas biológicos. Cada etapa debe documentar su contribución a homeostasis local y desempeño global.

---

1 Möbius strip - Wikipedia

[https://en.wikipedia.org/wiki/M%C3%B6bius\\_strip](https://en.wikipedia.org/wiki/M%C3%B6bius_strip)

2 Small world networks - Math Insight

[https://mathinsight.org/small\\_world\\_network](https://mathinsight.org/small_world_network)

3 4 GRAND: Graph Neural Diffusion

<https://arxiv.org/pdf/2106.10934>

5 [2006.06380] Pointer Graph Networks

<https://arxiv.org/abs/2006.06380>

6 7 [1603.08983] Adaptive Computation Time for Recurrent Neural Networks

<https://arxiv.org/abs/1603.08983>

8 [PDF] Riemannian Optimization for Deep Learning

<https://ntnuopen.ntnu.no/ntnu-xmlui/bitstream/handle/11250/3034256/no.ntnu:inspera:104646180:36318324.pdf?sequence=1>

9 Borromean rings - Wikipedia

[https://en.wikipedia.org/wiki/Borromean\\_rings](https://en.wikipedia.org/wiki/Borromean_rings)

10 Deep learning as Ricci flow | Scientific Reports

[https://www.nature.com/articles/s41598-024-74045-9?error=cookies\\_not\\_supported&code=84a34ef6-0c8e-409d-abd4-07b18f2285fa](https://www.nature.com/articles/s41598-024-74045-9?error=cookies_not_supported&code=84a34ef6-0c8e-409d-abd4-07b18f2285fa)