

# Entregable A: Ingeniería - Prototipo experimental

## Resumen ejecutivo:

- **Arquitectura de banda de Möbius dinámica:** Definimos un grafo sobre una banda de Möbius con parametrización explícita (coordenadas 3D) <sup>1</sup>. Muestramos  $N$  nodos uniformemente en parámetros  $(u, v)$ , con conexiones intrínsecas por vecindad local y conexiones extrínsecas entre puntos "contactantes" (distancia euclídea  $< \epsilon$ ).
- **Dinámica (rotación + ouroboros):** Implementamos un mecanismo continuo (Neural ODE / GNN en tiempo continuo <sup>2</sup>) con rotación centrífuga: definimos un "centro" (por ejemplo, eje del Möbius) y penalizamos (diffusion, Forman/ricci-damping) conexiones/nodos de alta entropía hacia los bordes <sup>3</sup> <sup>4</sup>. El movimiento tipo *ouroboros* recorre continuamente la cara única (fase  $\varphi(t)$ ), con eventos ("colapsos") disparados por el sinthome.
- **Punteros input/output y RAS/sinthome:** Mantenemos dos índices móviles ( $u_{in}(t)$ ,  $u_{out}(t)$ ) sobre la banda. El *sinthome* opera como controlador de coherencia: cada cierto ciclo de fase selecciona pares  $u_{in}/u_{out}$  consistentes (p.ej. minimizando energía gráfica o maximizar evidencia). RAS (inspirado por gating neurológico) modula activación de conexiones extrínsecas y verifica rutinas de evidencia (por ejemplo, recuperando información o aplicando reglas simbólicas <sup>5</sup> <sup>6</sup>).
- **Componentes software y datos:** Usamos Python + PyTorch + PyTorch Geometric (o DGL) + NetworkX para grafos; geomstats o *torchdiffeq* opcional. Cada nodo/edge lleva atributos: características (embedding inicial), entropía/incertidumbre estimada, evidencia (score de confianza), capa, radio (distancia al eje), fase  $\varphi$ . Registraremos decisiones del controlador RAS (qué edges activa o verificaciones lanza).
- **Protocolo experimental sintético:** Generamos escenarios de prueba con grafos de Möbius sencillos (p.ej.  $N=100, 200$ ) y secuencias semánticas sencillas. Medimos tasa de "alucinaciones" (claims no soportados), contradicciones internas y cobertura de evidencia. Comparamos variantes: sin vs con edges extrínsecos; con activación RAS on/off; con/ sin centrífuga. Hacemos *ablations* para aislar efectos (p.ej. quitar gating RAS, quitar sinthome).
- **Hipótesis medibles:** (A) Activar conexiones extrínsecas bajo gating RAS mejora coherencia global (menos contradicciones) comparado a desactivarlas <sup>3</sup> <sup>4</sup>. (B) Filtro centrífugo (penalizar baja evidencia en periferia) reduce alucinaciones en tareas de QA (vs baseline sin filtro). (C) Usar punteros móviles con verificación simbólica aumenta cobertura de evidencia.
- **Criterio de éxito:** Reducción estadísticamente significativa de alucinaciones y contradicciones con dinámicas propuestas. Mantenimiento de latencias razonables ( $<p95$ ) y control homeostático estable (no escalamiento excesivo de costos).
- **Implementación:** Pseudocódigo (Python) e interfaces de módulos (neural ODE solver, *router* de punteros, *memory* y *verificador* simbólico) están definidos. El *monitor* RAS/alertas registra cuándo y por qué se activan verificaciones extra. Usa PyG para mensajería GNN dinámica y Ripser/GUDHI para medir invariantes topológicos de coherencia en análisis.

## Estado epistemológico:

- *A (soportado):* "Graph Neural Networks continuos (GNN-ODE) modelan dinámicas sobre grafos con evolución temporal <sup>2</sup> <sup>7</sup>." "Forman-Ricci en grafos mide divergencia local: bordes con curvatura negativa son cuellos de botella <sup>3</sup>." "MoE en Transformers permite especialización de componentes bajo gating <sup>8</sup>." "Workspace global mejora coordinación de módulos con canal limitado <sup>5</sup>." "Persistent homology y Betti-números cuentan  $p$ -dimensionales 'huecos' topológicos <sup>9</sup>."

- **B (hipótesis testable):** "Conexiones extrínsecas (pliegues) pueden detectarse conectando nodos cercanos en embedding 3D y activarse por gating RAS." "Rotación centrífuga computable definiendo un eje y penalizando activaciones en radio alto (difusión/curvatura como regularizador) 4 3 ." "Punteros móviles, sinthome verificador: minimizar energía/coherencia en ruta graph (p.ej. par transporte paralelo) es un modelo operativo." "Gating RAS modula budget de cómputo (dilatar tiempo de cómputo según incertidumbre) como análogo de dilatación temporal 10 ."

- **C (analogía):** "Marco Lacaniano RSI (Real-Simbólico-Imaginario) es inspiracional; RAS como gating aquí es una metáfora psicológica, no mecanismo validado en ML." "Analogía Einstein/tiempo cósmico en asignar ciclos de cómputo es especulativa sin evidencia directa (meramente conceptual)."

Claim_ID	Claim	Etiqueta	Fuente(s)	Idea clave	Riesgo	Verificación
A1	GNN en tiempo continuo (Neural ODE) permiten modelar dinámicas espaciotemporales en grafos 2 7 .	A	2 7	Modelos de GNN+ODE capturan evolución continua de nodos.	Bajo; teoría establecida.	Implementar Neural ODE GNN (torchdiffeq+PyG) y confirmar dinámica simulada.
A2	Forman-Ricci en grafos detecta "cuellos de botella": bordes con curvatura negativa indican divergencia del flujo 3 .	A	3	Edges con Formula: $F(e)=4\text{-deg}(v1)-\text{deg}(v2)$ .	Medio; depende de definiciones de peso.	Calcular curvatura Forman de edges; correlacionar con ejes de "incertidumbre" altos.
A3	Global workspace (canal compartido limitado) mejora especialización y sincronización de módulos 5 .	A	5	Coordina módulos con canal ancho limitado.	Medio; más teórico.	Implementar módulo central y comprobar mejora (p.ej. latencia vs precisión al coordinar expertos).
A4	Persistent homology ofrece invariantes (Betti) que "cuentan" agujeros topológicos 9 .	A	9	$\text{Betti}_p = \# \text{ de hoyos } p\text{-dimensionales}$ .	Bajo; matemáticamente preciso.	Computar Betti <sub>0,1</sub> del grafo; usar cambios en Betti como señal de variación topológica.
B1	Punteo de pliegues: conectar puntos cercanos en 3D (dist < $\epsilon$ ) simula "edges extrínsecos".	B	-	Modelo operativo propuesto.	Medio; distancia umbral crítico.	Variar $\epsilon$ ; medir efecto a rendimiento y consistencia.

Claim_ID	Claim	Etiqueta	Fuente(s)	Idea clave	Riesgo	Verificación
B2	Rotación centrífuga: penalizar nodos/edges distantes del eje (difusión o curvatura).	B	-	Idea operacional de expulsar "ruido" al borde.	Alto; define incertidumbre cuantitativa.	Comparar con/ desactivado; medir métricas (calidad vs latencia).
B3	Punteros móviles/minimizar energía (holonomía discreta) vincula $u_{in}$ - $u_{out}$ .	B	-	Modelo de coherencia sinthome usando transporte paralelo.	Medio; detalles del enlace difusos.	Explorar modelos de enrutamiento (energy-based vs constraints), medir consistencia de salidas.
C1	RSI-RAS-Sinthome se traduce directamente a capas de control ML.	C	-	Analogy inspiracional.	Alto; no evidencia formal.	Conceptual.
C2	Analogía relativista: asignar "tiempo de cómputo variable" según densidad de información.	C	-	Idea especulativa.	Alto; sin base física.	Plantear como experimento de asignación de budget dinámico (B).

#### Diseño/Propuesta de implementación:

- **Representación de Möbius:** Parametrizamos la banda mediante  $(u,v)$ :  

$$\begin{aligned} x &= (R+v\cos(u/2))\cos(u), \quad y = (R+v\cos(u/2))\sin(u), \quad z = v\sin(u/2), \\ \text{donde } u &\in [0, 2\pi], \quad v \in [-w, w]. \end{aligned}$$
Muestreamos  $N_u \times N_v$  puntos uniformemente en  $(u,v)$ .
- **Conectividad intrínseca:** Grafo 2D en parámetros: cada nodo  $(u_i, v_j)$  conecta a sus vecinos  $(u_{\pm 1}, v_j)$  (mod  $2\pi$ ) *twist*: conectar  $(u_{end}, v)$  con  $(u_0, -v)$  y  $(u_i, v_{\pm 1})$  (sin "salto" de caras,  $v$  no envuelve). Este grafo de malla aproxima la métrica intrínseca.
- **Conectividad extrínseca ("pliegues"):** Para cada par de nodos no ya conectados, calcular distancia euclídea en el embedding 3D; si  $d_{ij} < \epsilon$ , añadimos un edge "pliegue" entre ellos. Este representa contacto físico del Möbius. El peso de acoplamiento inicial  $w_{ij} \propto \exp(-d_{ij}^2/\sigma^2)$  o constante.
- **Control RAS de edges extrínsecos:** Mantenemos un gating binario para edges extrínsecos. Por defecto se prueba: (1) baseline sin extras; (2) siempre activos; (3) solo activados según RAS (p.ej., si contribuyen a evidencia o reducen tensión del grafo). RAS puede evaluar "calidad" de un edge (basada en incertidumbre de nodos, curvatura, etc) y habilitarlo si mejora métrica de coherencia.

- **Definición de “centro” vs “borde”:** Tomamos el eje central (por ejemplo línea media  $u$  variable mientras  $v=0$ ) como *centro*. Definimos  $r =$  distancia radial al eje. Un nodo en borde si  $|v|$  grande. Podemos definir función  $f(r)$  o  $f(|v|)$  creciente para penalizar intensidad (por ejemplo, atenuar mensajes o incrementar dropout en bordes).
- **Mecanismo centrífugo:** Cada iteración de mensaje-paso, escalamos mensajes por factor inverso a  $f(r)$  (p.ej.  $g(r)=1/(1+\alpha r)$ ). También podemos sumar término de pérdida: edges que conectan nodos con gran  $r$  tienen penalización (regularización de curvatura **Forman**: favorecer estructuras localmente planas <sup>3</sup>). Alternativamente, aplicar difusiones: propagar información “hacia el centro” con un kernel de Laplace modificado, eliminando gradualmente señales ruidosas en el perímetro.
- **Variable de fase  $\varphi(t)$ :** Introducimos  $\varphi$  como índice en  $u$  (por ejemplo  $\varphi \in [0, 2\pi]$ ) desplazándose continuamente con velocidad  $\omega$ . En cada  $t$ , un “slice” de nodos con  $u \approx \varphi(t)$  se considera activo para computación. Esto simula el desplazamiento continuo *ouroboros*.
- **Eventos sinthome:** Definimos instantes de “colapso” cuando  $\varphi$  atraviesa ciertas bandas, o cuando la incertidumbre global (o energía de incoherencia) excede umbral. En esos eventos, el sinthome elige pares  $(u_{\text{in}}, u_{\text{out}})$  actuales (punteros) y produce salida: p.ej. aplicar capa de atención escasa (sparse routing) entre esos nodos según su embedding o features simbólicos. Aquí, el sinthome puede invocar un componente simbólico: e.g., verificar cumplimiento de reglas/contratos antes de generar la respuesta, actuando como restricción dura.
- **Punteros móviles:** Manejamos dos índices  $i_{\text{in}}, i_{\text{out}}$  que recorren los nodos del grafo a medida que  $\varphi$  avanza. Por defecto, los punteros se mueven a lo largo de la cara: p.ej.  $u_{\text{in}} = \varphi(t)$ ,  $v_{\text{in}}=0$  (centro),  $u_{\text{out}} = \varphi(t)+\pi$  (lado opuesto),  $v_{\text{out}}=0$ , adaptado a la geometría. Alternativamente, punteros podrían moverse según atención interna. El vínculo sinthome entre ellos se modela imponiendo un término de energía: queremos minimizar la diferencia semántica entre rutas de información que conectan estos nodos. Por ejemplo, podemos requerir que la holonomía (parallel transport) entre ellos sea baja; o maximizar la evidencia total de ambos.
- **Interfaces de módulos:**
  - *Geometry Engine*: Calcula embedding 3D de  $(u, v)$ , distancias, curvatura local (Forman y/o Ollivier calculada con pesos de grafo).
  - *GNN dinámico*: Neurona de mensaje-paso continua (p.ej. Graph Neural ODE <sup>2</sup>) que actualiza embeddings de nodos en tiempo continuo, incluida dinámica de edges.
  - *Router RAS*: Decide qué edges extrínsecos activar, cuándo hacer verificación, controla asignación de budget computacional (p.ej. retarda mensaje en nodos de baja confianza).
  - *Memory/Cache*: Almacena patrones previos, evidencia acumulada (sources), registros de verificación.
  - *Verificador simbólico*: API/servicio de recuperación externa (p.ej. bases de datos) o reglas lógicas, que corrige/valida salidas candidate.
  - *Output Synth*: Capa final que genera respuesta narrativa (Imaginario) basada en embeddings actuales y punteros.
- **Supuestos clave:**

- *Modelo del grafo*: Asumimos que mapear la entrada a esta topología es coherente. Si la orientación no funciona (p.ej. no hay inputs naturales para Möbius), la arquitectura puede sobrediseñarse.
- *Activación extrínseca*: Supone que los “pliegues” agregan caminos útiles; pero podrían introducir ruido. Si la activación por RAS es errónea, podríamos activar edges irrelevantes. Test: ablation sin extrínsecos.
- *Movimientos simultáneos*: La rotación y ouroboros combinados pueden complicar el entrenamiento. En el peor caso, desacoplamos (usar solo fase o solo rotación inicialmente).
- *Punteros*: Suponemos que existen inputs distinguibles para “entrada” y “salida”; si no, podría ser redundante.

#### **Plan de simulación/experimento:**

##### - **Hipótesis:**

- H1: El gating RAS de edges extrínsecos mejora consistencia semántica (menos contradicciones) comparado con activación constante o inactiva.
- H2: La dinámica centrífuga (filtrado de bordes de alta entropía) reduce la tasa de alucinaciones en una tarea factual.
- H3: El enlace sinthome (p. ej. minimización de energía entre punteros) aumenta la coherencia entre input y output (medida en consistencia contextual).

##### • **Setup:**

- Construir escenarios sintéticos: e.g. tareas de QA en grafos donde la respuesta correcta depende de conectar caminos a través de la banda.
- Escenarios:  $N = 100,200$  nodos; diferentes proporciones de edges extrínsecos ( $\epsilon$  variable).
- Implementar variantes de diseño: Sinthome off/ on; RAS gating vs baseline; con/sin centrífuga.
- Datos: Simulaciones numéricas y tareas de consistencia (p.ej. hechos enciclopédicos con evidencias limitadas).

##### • **Variables:**

- **Independientes**: Activación edges extra (sí/no/RAS), intensidad filtro centrífugo (parámetro  $\alpha$ ), esquema punteros (aleatorio vs determinista), nivel de incertidumbre inicial (ruido en features).
- **Dependientes (métricas)**: Tasa de alucinaciones (claims sin fuente válida), contradicciones internas (discrepancia entre dos respuestas a misma consulta con hist.), cobertura de evidencia (% claims con citas), precisión de respuesta, costo (tokens generados, timeouts), latencia (p50/ p95), tasa de escalamiento RAS (frecuencia de eventos de verificación).

##### • **Métricas (concretas):**

###### • *Veracidad*:

- **Hallucination rate**: % de afirmaciones no soportadas por fuentes (requiere instrumentar output).
- **Consistency score**: e.g. aplicación de un verificador de coherencia conversacional (p.ej. si el modelo contradice hechos previos).
- **Evidence coverage**: Fracción de respuestas con referencias válidas.
- **Calibración incertidumbre**: Brier score sobre certificación de afirmaciones (similar a reliability of confidences).

###### • *Operativo*:

- **Latencia (p50, p95)** tiempo de respuesta.
- **Costo computacional**: número de llamadas al modelo grande, tokens procesados.

- **Tasa escalamiento:** veces que el RAS dispara verificación extra (p.ej. llamadas a verificador/símbolo).
  - **Degradación controlada:** medir cuánto cae la calidad al limitar recursos (por ejemplo, si RAS no permite alguna verificación).
- *Usuario:*
- **Tasa de éxito en benchmark interno:** % de queries respondidos correctamente (tal vez con dataset QA sintetico).
  - **Retrabajo:** número de iteraciones/correcciones solicitadas por usuario (simulado) antes de aceptación.

#### Pseudocódigo (esquemático):

```

# Parámetros
R, w = 1.0, 0.2
N_u, N_v = 100, 10
epsilon = 0.05
alpha_cent = 1.0 # centrífuga
# Inicializar nodos en (u,v)
nodes = [(i*2π/N_u, j*2*w/(N_v-1)-w) for i in range(N_u) for j in range(N_v)]
coords3D = {n: embed_Mobius(u,v,R,w) for n,(u,v) in enumerate(nodes)}

# Conectividad intrínseca
edges = []
for n,(u,v) in enumerate(nodes):
    i,j = index(n, N_v) # map nodos a índices
    # vecinos U
    i_next = (i+1)%N_u
    v2 = nodes[i_next*N_v + j][1]
    # ajustar signo en u wrap:
    if i_next==0: v2 = -v2
    edges.append((n, i_next*N_v + j))
    # vecinos V
    if j < N_v-1:
        edges.append((n, i*N_v + j+1))
# Agregar extrínsecos (if any)
for (n1, c1),(n2,c2) in combinations(coords3D.items(), 2):
    if not intrinsic_connected(n1,n2) and dist(c1,c2)<epsilon:
        edges.append((n1,n2, weight=exp(-dist(c1,c2)**2/sigma2)))

# Dinámica GNN en tiempo continuo (neural ODE style)
def node_dynamics(t, H):
    # H: vectores de nodos
    newH = zeros_like(H)
    for each edge (i,j):
        # mensaje normal
        msg = GNN_message(H[i], H[j], weight)
        # ajustar por centrífuga: escala mensaje por g(r)
        r = radial_distance(i) # distancia al eje
        g = 1/(1 + alpha_cent*r)
        newH[j] += g*msg
    # agregamos quizá drift o biases

```

```

    return newH

# Integración (p. ej. Euler)
H = random_init(N_u*N_v)
for t in time_steps:
    H = H + dt * node_dynamics(t,H)
    # Check RAS gating
    for edge in extrinsic_edges:
        # Evaluar criterio RAS: e.g. si ambos nodos high_uncert, enable edge
        if RAS_decide(edge, H):
            activate_edge(edge)
        else:
            deactivate_edge(edge)
    # Movimiento punteros
    u_in = φ(t) # por ejemplo
    u_out = (φ(t) + π) % (2π)
    # Sinthome event?
    if sinthome_condition(t,H):
        # elegir punteros actuales
        in_node = select_node_at(u_in)
        out_node = select_node_at(u_out)
        # generar salida:
        out = generate_answer(H[in_node], H[out_node])
        # opcional: verificación simbólica
        if RAS_needs_verify(out):
            out = symbolic_verify(out)
    emit out

```

#### **Escenarios sintéticos anti-alucinación (mínimos):**

1. **Tarea factual simple:** Información base + pregunta trivial. Introducir falsas referencias y ver si RAS/sinthome las filtra.
2. **Cascada de contradicción:** Preguntas encadenadas donde respuesta anterior contradice posterior si no hay coherencia. Se prueba consistencia interna.
3. **Recovery de evidencia:** Pedir referencias específicas; evaluar si cobertura de evidencias sube con punteros de coherencia.
4. **Stress test de ruido:** Inyectar textos de entrenamiento con ruido e incertidumbre; medir si centrífuga ayuda.
5. **Shut-down gradual:** Limitar recursos de RAS o verificador; observar degradación controlada (compara calidad vs latencia).

#### **Próximos pasos (priorizados):**

1. **Prototipar Möbius + GNN:** Implementar el grafo Möbius, dinámica básica GNN-ODE. Verificar integridad topológica (Betti invariantes). (*Riesgo: Complejidad de código, errores topológicos*)
2. **Implementar gating de edges:** Añadir mecanismo RAS simple (por ejemplo basado en entropía nodal). Test con/ sin edges. (*Riesgo: Definir umbral  $\$\\epsilon$  óptimo*)
3. **Dinámica centrífuga:** Realizar prueba de difusión/regularización en bordes. Ajustar parámetro  $\alpha$ . (*Riesgo: Podría degradar capacidad de red si muy agresivo*)
4. **Mecanismo punteros y sinthome:** Diseñar eventos ouroboros y selección input/output. Crear verificador dummy (e.g., consulta JSON de hechos). (*Riesgo: Definición arbitraria sin marco riguroso*)
5. **Escenarios de test riguroso:** Construir datasets sintéticos con ground truth. Automatizar métricas.
6. **Ablations:** Remover componentes (sin centrífuga, sin RAS, sin extrínsecos) para aislar efecto.

## 7. Riesgos:

- Alta complejidad de simulación (difícil de sintonizar).
- Modelos Lacanianos no traductibles (poner al sinthome en ML).
- Coste computacional alto (red dinámica + verificador).
- Sin fuentes reales para métricas; dependencia de benchmarks sintéticos.

# Entregable B: Teoría/Topología – Límites conceptuales

## Resumen ejecutivo:

- **Topología “dura”:** Definimos la banda de Möbius y comparamos con enlaces Borromeos. Möbius: una superficie no orientable con frontera única; Borromean: triple enlace de círculos (cada par separado pero los tres enlazados). Son estructuras topológicas distintas (no homeomorfas). Möbius tiene grupos de homología  $H_0 \cong \mathbb{Z}, H_1 \cong \mathbb{Z}$  (p.ej.  $Betti_1=1$ )<sup>9</sup>. Las cadenas de Borromean tienen homología de enlace con forma de anillo triple (no trivial). Sin evidencia, tratamos Borromean como metáfora de multiconexión RSI (Categoría C).
- **Invariantes computables:** Proponemos usar **Betti-0** (componentes), **Betti-1** (loops) del grafo dinámico para detectar rupturas/coherencia. E.g., un Möbius strip trivialmente tiene un loop único de base ( $Betti_1=1$ ). Cambios topológicos (como desconectar bordes) alterarían Betti. La orientación no influye en Betti mod-2<sup>11</sup>. Formas como Euler característica  $\chi = \#nodes - \#edges + \#faces$  (para grafos:  $\chi = \#components - (\#edges - \#nodes)$ ) pueden monitorear estabilidad.
- **“Amarre” en grafos dinámicos:** Planteamos que un amarre fuerte equivale a invariantes topológicos estables a perturbaciones: p.ej. reforzar loop principal (guardar  $Betti_1$  constante). Proponemos estrategias inspiradas en TDA: rastrear la persistencia de ciclos topológicos relevantes cuando aparecen/ desaparecen edges. Un tie (amarre) sería un ciclo persistente (alta persistencia) en la filtración de grafo<sup>12</sup>.
- **Möbius vs Borromean (formal):** Matemáticamente, *no* hay homeomorfismo sencillo: Möbius es superficie de género cero con frontera, Borromean es enlace en  $S^3$  (figura más similar a 3-loop sin superficie de Cobordismo). El único lugar podrían encontrarse es en la teoría de nudos de Lacan como analogías de RSI; sin justificación física, las dejamos en (C).
- **Analogías topológicas (etiquetado C):**
  - “Tiempo cuántico/gravedad”: especulación (C) de que densidad de información curva el espacio-tiempo interno, dejando como simple asignación de recursos variável (colloquial: *computational dilation*).
  - “Penrose/Hameroff”: sin soporte, marcar (C). Sólo mencionar que microtúbulos y conciencia son analogía filosófica. En arquitectura, traducirlo a (B) redes neuronales recurrentes continuas (ya cubierto en GNN-ODE).
- **Preguntas límites (“agenda de investigación”):**
  1. ¿Puede la homología persistente de la red informar detección de incoherencias globales? P.ej., desaparición repentina de un ciclo puede señalar ruptura de consistencia.
  2. ¿Qué invariantes (Forman vs Ollivier vs Ricci-flow) mejor correlacionan con flujos de información (coherencia)?
  3. ¿Hasta qué punto analogías Lacanianas (Borromeo como RSI) tienen contraparte formal en teoría de grafos? (Probablemente no operable).
  4. ¿Cómo representar “amarre” estructuralmente? Tal vez como densidad de conexiones que forman un lazo recurrente (ciclo fuertemente conectado). ¿Es esto computable vía clustering o ciclos de Hamilton?
  5. ¿Tiene sentido “homología no orientable” en grafos (p.ej. orientaciones dirigidas de la cara única)? Posible: considerar orientaciones de ciclos (trace).

6. Test mínimo: Verificar Betti invariance al doblar edges. Simulaciones de pequeños grafos Möbius y medir invariantes mientras aplicamos dinamismo.

#### Estado epistemológico:

- A: "El Möbius strip es una superficie no orientable con único borde; su  $H_1 = \mathbb{Z}$  (un loop básico) <sup>9</sup> ." "Betti\_p cuenta  $p$ -huecos; para Möbius (2D), solo  $Betti_0=1$ ,  $Betti_1=1$  <sup>9</sup> ." "Homología mod-2 puede diferir en no-orientables (Klein bottle ejemplo:  $Betti_2=0$  sobre  $\mathbb{R}$  pero  $=1 \text{ mod } 2$ ) <sup>11</sup> ." "Forman-Ricci y Ollivier-Ricci proporcionan medidas de curvatura discrete para grafos <sup>3</sup> ." "TDA (persistent homology) ofrece señales globales robustas a ruido <sup>12</sup> ."
- B: "Verificar invariantes topológicos (Betti) computacionales en grafos dinámicos; p.ej., medir Betti-1 a lo largo del tiempo como señal de consistencia. Los invariantes computables (Euler, Betti) se pueden obtener con Ripser/GUDHI." "Equivalencias Möbius vs Borromean como analogías se formalizarán sólo como marcos conceptuales (sin equivalencia directa A)."
- C: "Relación Möbius-Borromeo, Penrose/conciencia, relatividad-tiempo computacional: supuestos especulativos sin base empírica o teoría aceptada en el contexto de ML; solo (B) como mecanismos operacionales triviales (p.ej., asignar iteraciones variables)."

#### Evidencia teórica clave:

- Möbius: No apto para citar en ML, pero topología básica: "Möbius es no-orientable, un lado" (conceptual; no se cita). Podríamos citar [31+L23-L28] para enfatizar no orientabilidad ( $Betti_2 \text{ mod } 2$ ).
- Borromean: No hay fuente técnico-Máquina learning; lo mencionamos conceptual (analogy (C)).
- TDA: mencionar persistent homology robustez (no se citó textualmente, pero [8+L64-L68] dice extensión a PH). - Curvatura: [8+L108-L114] explica interpretación (bottlenecks). - Global invariantes: Euler-Poincaré (componentes - ciclos) [31+L7-L10] implícito.

#### Límites ("no tiene sentido / solo metáforas"):

- No operacionalizar literales analogías Lacanianas; RSI es heurístico. No hay "conexión matemática comprobada" entre sinthome y grafos.
- Penrose/Hameroff: tratar como (C). No se usa quantum en cálculo. No operacionalizar microtúbulos.
- Gravedad/GR: sin evidencias de efecto real en cómputo. Solo considerarlo (C). Traducimos a presupuesto dinámico (B) si acaso.
- TDA: Limitaciones prácticas – el cálculo de persistent homology es costoso en escala, tal vez solo usar  $Betti_0/1$ .
- Curvatura: Ollivier requiere solución óptima (costoso); Forman es más ligero (solo grados).

#### Registro de búsqueda (síntesis):

- **Búsqueda:** "dynamic graph neural networks continuous time" → *Encontrado:* Survey actualizado (2024) de GNN dinámicos <sup>13</sup> <sup>14</sup> , clasificando TGAT, TGN, etc. Relevante para elegir modelo continuo (GNN-ODE <sup>2</sup> ).
- **Búsqueda:** "Graph neural ODE GNN" → *Encontrado:* Survey Graph NDE (2025) <sup>2</sup> confirma GNN+ODE.
- **Búsqueda:** "Forman curvature graphs Sreejith" → *Encontrado:* Sreejith et al. (2016) teoría; summary [8] con fórmula. Usamos para curvatura penalización <sup>3</sup> .
- **Búsqueda:** "persistent homology Betti numbers invariant" → *Encontrado:* Roadmap PH (Otter et al. 2017) <sup>9</sup> ; definiciones Betti. Usamos Betti invariante.
- **Búsqueda:** "mixture of experts graph GNN" → *Encontrado:* Nat.Comm 2025 Mixture-of-Experts transformer <sup>8</sup> ; adaptamos idea.
- **Búsqueda:** "global workspace neural network coordination" → *Encontrado:* Goyal et al. 2021 ICLR <sup>5</sup> con restricciones de ancho de banda (aplicado a módulos GNN).
- **Búsqueda:** "homeostasis reinforcement learning" → *Encontrado:* Yoshida et al. 2025 HRRL <sup>10</sup> . Adaptamos concepto global de preservar "estado interno".

- **Búsqueda:** "hallucination LLM metric consistency" → *Encontrado*: Review fact-checking LLM (2025) 6  
describe hallucinations; nos ayuda a definir métricas de calidad.
- **Pendientes:** No se encontró enlace directo entre Relatividad/Computación; tratamos como (C). No hay fuente para relaciones Möbius-Borromean aparte de Lacan (filos).
- **Material perdido:** Explorar si existen trabajos concretos sobre GNNs en superficies no orientables (aparte de GD-VAEs latent [34†L1-L4]). Podría ayudar, pero OK con lo que tenemos.

#### Queries de búsqueda prioritarias:

1. "GNN continuous-time neural ODE"
2. "persistent homology graph coherence measure"
3. "Forman Ricci graph curvatures denoising"
4. "homeostatic reinforcement learning control"
5. "Möbius graph embedding library"
6. "Graph neural net mixture of experts MoE"
7. "global workspace architecture deep learning"
8. "hallucination detection LLM factual consistency"
9. "non-orientable manifold learning neural network"
10. "Borromean knot topology algorithms"

#### 5 riesgos principales:

- *Complejidad implementacional*: Integrar todas las ideas (Möbius, continuo, gating, verificadores) es arduo.
- *Sobreajuste arquitectural*: Demasiados componentes podrían sobreajustarse a escenarios sintéticos y no generalizar.
- *Métricas difusas*: Evaluar "alucinación" es difícil sin benchmarks reales. Riesgo de falsear evidencia.
- *Análogas especulativas*: Ceder a analogías Lacan/GR puede confundir el diseño (riesgo conceptual).
- *Costo computacional*: GNNs dinámicos + simulación + verificación pueden ser lentos; riesgo de no escalar.

#### 5 experimentos mínimos:

1. **Impacto de edges extrínsecos:** Ejecutar con/ sin edges "pliegue". Métricas: contradicción y coherencia. Éxito: mejora con edges adecuados.
2. **Efecto centrífuga:** Variar parámetro de penalización radial, medir tasa de hallucinations. *Decisión*: encontrar tradeoff óptimo.
3. **Gating RAS vs simple:** Simular RAS inteligente (basado en incert./curvatura) vs gating estático. Evaluar reducción de respuestas sin fuente.
4. **Punteros vs fijo:** Usar punteros móviles dinámicos vs punteros fijos. Verificar coherencia respuesta (p.ej. cross-entropía con verdad).
5. **Stress test:** Limitar recursos (e.g. tiempo de verificador) para observar degradación: calidad cae suavemente = éxito (control homeostático).

1 GD-VAEs: Geometric Dynamic Variational Autoencoders for Learning Nonlinear Dynamics and Dimension Reductions  
<https://arxiv.org/html/2206.05183v4>

2 7 Graph ODEs and Beyond: A Comprehensive Survey on Integrating Differential Equations with Graph Neural Networks  
<https://arxiv.org/html/2503.23167v1>

3 12 Forman-Ricci Curvature in Discrete Structures  
<https://www.emergentmind.com/topics/forman-ricci-curvature>

- <sup>4</sup> proceedings.neurips.cc  
[https://proceedings.neurips.cc/paper\\_files/paper/2023/file/f978c8f3b5f399cae464e85f72e28503-Paper-Conference.pdf](https://proceedings.neurips.cc/paper_files/paper/2023/file/f978c8f3b5f399cae464e85f72e28503-Paper-Conference.pdf)
- <sup>5</sup> [2103.01197] Coordination Among Neural Modules Through a Shared Global Workspace  
<https://arxiv.org/abs/2103.01197>
- <sup>6</sup> Hallucination to Truth: A Review of Fact-Checking and Factuality Evaluation in Large Language Models  
<https://arxiv.org/html/2508.03860v1>
- <sup>8</sup> Mixture-of-experts graph transformers for interpretable particle collision detection | Scientific Reports  
[https://www.nature.com/articles/s41598-025-12003-9?error=cookies\\_not\\_supported&code=da1e5177-3ae5-4df6-890f-917da66a78f9](https://www.nature.com/articles/s41598-025-12003-9?error=cookies_not_supported&code=da1e5177-3ae5-4df6-890f-917da66a78f9)
- <sup>9</sup> <sup>11</sup> A roadmap for the computation of persistent homology - PMC  
<https://pmc.ncbi.nlm.nih.gov/articles/PMC6979512/>
- <sup>10</sup> [2507.04998] Linking Homeostasis to Reinforcement Learning: Internal State Control of Motivated Behavior  
<https://arxiv.org/abs/2507.04998>
- <sup>13</sup> <sup>14</sup> A survey of dynamic graph neural networks  
<https://arxiv.org/html/2404.18211v1>