

Paso 3 – Diseño de Benchmark y Experimento

1. Resumen ejecutivo

Para validar el efecto de **Forman-Ricci cleaning** en la memoria/grafos del agente y el beneficio del **gating entrópico**, diseñamos un benchmark sintético reproducible. El objetivo es medir cómo la eliminación de aristas de curvatura baja mejora el enrutamiento de memoria y la robustez ante ruido, sin añadir latencia excesiva. Utilizamos una mezcla de grafos sintéticos con comunidades y ruido controlado y, en una segunda fase, subgrafos reales de tus memorias/diario cuando estén disponibles.

2. Hipótesis operativas

1. **H1 – Denoising estructural:** al aplicar la curvatura de Forman-Ricci y podar aristas con curvatura negativa alta (o peso estirado), se eliminan aristas que poco aportan a la propagación de información; los caminos sobre el grafo resultan más cortos y aumentan la probabilidad de llegar al nodo objetivo. La literatura muestra que aristas con **curvatura baja** ofrecen poca contribución a la representación de un nodo y pueden eliminarse sin degradar su información ¹.
2. **H2 – Gating inteligente:** un RAS que active el cleaning sólo cuando la entropía/inseguridad supera un umbral mantiene el beneficio de H1 con menor coste de cómputo, comparado con limpiar en todos los turnos.

3. Diseño del dataset sintético

3.1 Grafo base

- **Modelo:** Generamos comunidades con un **Stochastic Block Model (SBM)** o gráfico de **planted partitions**, para simular módulos temáticos del agente. Cada comunidad contiene un número fijo de nodos (por ejemplo 30–100) con alta probabilidad de conexión intra-comunidad y baja probabilidad inter-comunidad.
- **Tipos de nodo:** replicamos las clases del MVP (`MEMORY_CHUNK`, `TOOL`, `EXPERT`), pero para el benchmark sintético basta con nodos sin atributos; lo relevante son las estructuras.
- **Aristas y pesos:** inicialmente, asignamos un peso base w_{uv} (distancia/coste), por ejemplo $w = 1$ para aristas intra-comunidad y $w = 2$ para inter-comunidad. La semántica de peso como distancia permite interpretar el cleaning como la reducción de ruido y la poda de “caminos largos/no informativos”.

3.2 Inyección de ruido

Para evaluar la resiliencia, añadimos **aristas espurias**:

1. **Aleatorias:** se conectan nodos aleatoriamente con probabilidad p_{noise} (por ejemplo 1–10 % de los pares), con pesos bajos. Estas aristas representan correlaciones espurias o saltos de memoria inadecuados.

2. **Adversarias:** se conectan nodos de comunidades muy lejanas con pesos similares a las aristas legítimas. Este ruido afecta más el enrutamiento, pues confunde al algoritmo de top-k.

3.3 Variación de parámetros

- **Tamaño de las comunidades:** 30, 50, 100 nodos.
- **Densidad intra/inter:** probabilidad de arista dentro/between communities (por ejemplo 0.2 intra, 0.02 inter).
- **Ruido:** porcentajes de aristas espurias (0 %, 5 %, 10 %, 20 %).
- **Parámetros de cleaning:** pasos t , coeficiente η , umbral de poda w_{max} (percentil 90–95 del peso actualizado).
- **Umbral de gating:** valores de entropía o “inseguridad” que disparan limpieza (por ejemplo entropía > 0.6). Estos umbrales se calibran en pruebas previas.

4. Procedimiento experimental

1. **Construcción del grafo:** para cada configuración generamos n_{graphs} instancias (por ejemplo 10) para robustez estadística.
2. **Inyección de ruido:** agregamos aristas espurias según p_{noise} .
3. **Cálculo de curvatura:** calculamos la curvatura de Forman seleccionada (simple o AFRC) para cada arista, repitiendo t pasos de Ricci-flow y actualizando pesos según la regla estirar/contraer.
4. **Poda:** eliminamos aristas cuya distancia/curvatura supera el umbral w_{max} (es decir, aristas con baja información que se estiran y se vuelven costosas). Según la literatura, estas aristas “de curvatura baja” aportan poco a la representación y su eliminación reduce la complejidad sin perder información ¹.
5. **Enrutamiento:** para cada par origen-destino dentro de la misma comunidad (o para un conjunto aleatorio de consultas), calculamos caminos por top- k y registramos si el destino deseado aparece entre los k primeros caminos/nodos.
6. **Comparación con baselines:** repetimos los pasos 3–5 **sin cleaning** y **con limpieza continua** para comparar con el cleaning con gating.
7. **Medición del gating:** introducimos un proxy de **entropía** en el agente (por ejemplo entropía de la distribución de similitud de embeddings) y limpiamos sólo cuando este valor supera el umbral. Registramos cuántas veces se activa el cleaning y el tiempo de ejecución.

5. Métricas de evaluación

1. **Precisión de enrutamiento (accuracy@k):** proporción de consultas en las que el nodo objetivo aparece entre los k mejores candidatos. Con cleaning se espera mayor precisión que con grafo sucio.
2. **Edge precision/recall:** si guardamos el grafo “verdadero” sin ruido, medimos la fracción de aristas espurias eliminadas (recall) y la fracción de aristas legítimas preservadas (precision). La teoría de Ricci indica que las aristas con **baja curvatura** son candidatas a eliminar ¹.
3. **Longitud de camino promedio:** número medio de saltos entre nodos origen-destino. El cleaning debería acortar caminos al eliminar rutas ruidosas.
4. **Costo computacional:** tiempo de cálculo y latencia del pipeline, tanto con cleaning continuo como con gating. Esto evalúa H2.

6. Criterios de éxito

La hipótesis H1 se considera validada si, para un k fijo, la precisión de enrutamiento en grafos limpiados supera en al menos X % (por ejemplo 10 %) la del baseline sin limpiar, manteniendo o reduciendo la longitud de camino. H2 se cumple si la precisión del cleaning con gating es comparable a la del cleaning continuo, pero con reducción significativa del coste (por ejemplo reducción ≥ 30 % de tiempo o número de limpiezas).

7. Consideraciones y riesgos

1. **Modelo de ruido:** grafos sintéticos no capturan toda la complejidad de una memoria real; se deben validar en subgrafos de tu sistema cuando estén disponibles.
2. **Estimación de entropía:** la definición de entropía/inseguridad para disparar cleaning puede variar (por ejemplo enrutamiento, similitud de embeddings). Este benchmark permite experimentar con diferentes proxies.
3. **Curvatura aproximada:** la curvatura de Forman puede ampliarse con términos de triángulos (AFRC). La literatura sugiere que el uso de **curvatura aumentada** con triángulos mejora la robustez manteniendo coste casi lineal ².

8. Próximos pasos

Después de validar el benchmark sintético, se deben aplicar los mismos experimentos en subgrafos extraídos de tus memorias conversacionales o bases de conocimiento reales para evaluar la efectividad en un contexto operativo.

¹ tnnls-23.pdf

<https://songdj.github.io/publication/tnnls-23/tnnls-23.pdf>

² Augmented Forman-Ricci Curvature

<https://www.emergentmind.com/topics/augmented-forman-ricci-curvature-afrc>