



Actualización del plan vNext: RAS-Sinthome anti-alucinación y prior Möbius-Borromeo

Marco conceptual: sinthome como operador de anudamiento y control de coherencia

Tu ampliación tiene un anclaje conceptual muy claro en el uso lacaniano de la topología: en **Seminario XXIII (Le Sinthome)**, Lacan trabaja explícitamente con el **anudamiento borromeo** y con la idea de un “cuarto” elemento (el *sinthome*) que puede sostener el enlace cuando el amarre entre **Real, Simbólico e Imaginario (RSI)** no “cierra” de forma estable. ¹ En la literatura secundaria lacaniana (y en lecturas clínicas) se repite la formulación: el cuarto término “asegura” la relación entre los tres registros al **tejerse** de modo que produzca un **borromeo de cuatro**, es decir, una estructura globalmente estable a pesar de fragilidades locales. ²

Para ingeniería, la traducción útil no es “topología como metáfora” sino **topología como especificación de fallo**: la alucinación se puede operacionalizar como (i) **pérdida de anclaje** (*fall de amarre* entre módulos/estados/evidencia) y (ii) **mala postura epistémica** (*calibración defectuosa*: sobreconfianza, o incapacidad de abstenerse). Esta separación es consistente con taxonomías modernas de alucinación en LLMs que distinguen entre **factuality hallucination** (desalineación con hechos verificables) y **faithfulness hallucination** (desalineación con la entrada, el contexto o la fuente disponible), donde las contradicciones inter-turno y respecto al contexto caen naturalmente en “**faithfulness**”. ³

Dos aclaraciones que fortalecen tu “eje B” sin perder rigor:

- **Borromeo** (en teoría de nudos/enlaces) pertenece a la familia de los **Brunnian links**: un enlace no trivial que se vuelve trivial al remover cualquier componente. ⁴
- **Möbius** es una **superficie no orientable**; no es “lo mismo” que Borromeo, pero sí puede funcionar como **soporte geométrico o prior de conectividad** para que el sistema tenga “atajos” coherentes (lo que tú llamas pliegues/arrugado), y luego imponer un “test de integridad” tipo borromeo sobre flujos/campos. ⁵

En síntesis: tu **RAS-Sinthome** es un *controlador de activación + verificación* que se comporta como “cuarto anillo” (no porque imite clínica, sino porque implementa una **función de anudamiento**: impedir que el sistema produzca salida cuando el enlace entre evidencia-estado-respuesta está roto o mal calibrado). ⁶

Señales interoceptivas para riesgo de alucinación

Tu propuesta de cuatro señales (`epistemic_uncertainty`, `grounding_gap`, `consistency_drift`, `hallucination_risk`) está alineada con una tendencia fuerte: **medir incertidumbre, detectar falta de soporte y aplicar verificación/abstención en función de riesgo**. En particular, hay evidencia de que la “confianza” de modelos modernos puede estar **mal calibrada** (en redes neuronales en general), y se han propuesto métodos específicos y *surveys* para **confidence estimation y calibration** en LLMs. ⁷

A continuación, una operacionalización concreta (con estimadores típicos) que mantiene tu intención, pero reduce ambigüedad de implementación.

epistemic_uncertainty (**incertidumbre/calibración**)

En LLMs, la incertidumbre útil suele aproximarse mediante señales como: entropía de distribución de tokens, márgenes logit/probabilidad, variabilidad bajo muestreo (temperature / nucleus), o **disagreement** entre múltiples muestras/"expertos" (ensembling). Dos piezas de evidencia muy directamente aprovechables:

- **Self-Consistency**: muestrear múltiples cadenas de razonamiento y elegir la respuesta más consistente mejora desempeño en razonamiento; su lógica ("si hay múltiples rutas, usa consenso") formaliza una señal de *disagreement* explotable como incertidumbre. ⁸
- **SelfCheckGPT**: para detección de alucinación "caja negra", la divergencia/contradicción entre muestras ante el mismo prompt se usa como proxy: si el modelo "sabe", las muestras tienden a ser estables; si está inventando, divergen. ⁹

Como pieza de calibración "clásica" (más de clasificación que de generación, pero útil como base conceptual), **temperature scaling** se destaca como técnica simple y efectiva para mejorar calibración en redes modernas. ¹⁰

En tu arquitectura, esto sugiere que **epistemic_uncertainty** debería componerse con al menos dos familias: (a) **incertidumbre intrínseca** (entropía/margen) y (b) **incertidumbre por desacuerdo** (varianza entre muestras/expertos).

grounding_gap (**brecha salida-evidencia**)

Aquí conviene distinguir *dos casos*, porque cambian los falsos positivos:

- Cuando **hay evidencia disponible** (RAG, archivos, workspace, memoria corta), el "gap" es estrictamente: *¿cuánto de lo que afirmo está sustentado por lo recuperado/permitido?*
- Cuando **no hay evidencia disponible**, el "gap" deja de ser un fallo del sistema y pasa a ser presión para: (i) recuperar, (ii) abstenerse, o (iii) responder con hipótesis explícitamente marcadas como tales.

Dos referencias "de anclaje" para tu métrica: * **RAG** formaliza el patrón de combinar memoria paramétrica con memoria no paramétrica recuperada, y reporta que puede producir lenguaje más factual/específico que un modelo sólo paramétrico en tareas intensivas en conocimiento. ¹¹

FactScore* propone evaluar factualidad descomponiendo la generación en "hechos atómicos" y midiendo el porcentaje apoyado por una fuente confiable (vía retrieval + verificación), precisamente el tipo de contabilidad que necesitas para "tasa de afirmaciones sin soporte". ¹²

consistency_drift (**deriva/contradicción respecto a estado y turnos previos**)

En taxonomías recientes, la alucinación también incluye **contradicciones** con el contexto o con lo ya dicho (faithfulness issues). ³ Para operacionalizarlo en un sistema conversacional, suele bastar con: * extraer "compromisos" (claims/decisiones) de los últimos k turnos y del estado actual, * y correr detección de contradicción (NLI u otro verificador) **sólo sobre esos compromisos** para evitar el costo de comparar con todo.

Esto encaja muy bien con tu idea de "amarre": cuando el sistema se contradice, no es sólo un error factual, sino un fallo del **mecanismo que mantiene coherencia global**.

hallucination_risk (**score compuesto**)

Aquí el punto clave es evitar que sea "una suma arbitraria". Un diseño robusto usa

`hallucination_risk` como **variable de decisión** (policy input) y evalúa su calidad como **clasificador de dificultad/riesgo** (calibración de riesgo). Las literaturas de **abstención** y “know your limits” resaltan que cuando el modelo puede abstenerse, se debe evaluar la relación riesgo-cobertura (no sólo accuracy). ¹³

Políticas de escalamiento y verificación

Tu política RAS (bajo/medio/alto riesgo) coincide con dos líneas de investigación “ingenierizables”:

- **Cómputo condicional / enrutamiento:** activar “subredes” o rutas distintas según el input (Mixture-of-Experts), donde un router elige expertos por ejemplo. ¹⁴
- **Cómputo adaptativo:** decidir cuántos pasos de cómputo ejecutar según dificultad (Adaptive Computation Time). ¹⁵

Leyendo tu RAS como “router + controlador de profundidad”, la política se vuelve el equivalente funcional de **asignar más cómputo** (y más verificación) a casos más inciertos, lo cual ya tiene precedentes en el diseño de sistemas adaptativos. ¹⁶

A nivel de “cinturón de verificación”, tu tríada CoVe/PRM/DSVD está muy bien elegida porque cubre **tres estilos complementarios**:

CoVe como verificación por preguntas internas y desacople de sesgo del borrador

Chain-of-Verification (CoVe) propone un pipeline en el que el modelo (i) redacta, (ii) genera preguntas de verificación, (iii) responde esas preguntas de forma más independiente para evitar que el borrador sesgue, y (iv) produce una respuesta final verificada, reportando reducción de alucinaciones en varios setting (incluyendo preguntas tipo lista y QA closed-book). ¹⁷

PRM como supervisión/verificación del proceso (paso a paso)

Los **Process Reward Models** buscan puntuar/verificar **pasos intermedios**, mitigando errores en procesos de razonamiento (en especial matemático), pero con advertencias metodológicas: su construcción y evaluación es difícil, hay sesgos en Best-of-N y problemas en generación de datos/etiquetado; esto es relevante para tu diseño porque te obliga a definir muy bien “qué significa paso correcto” en tu pipeline de verificación. ¹⁸

DSVD como verificación en tiempo de decodificación (corrección dinámica)

DSVD (“Dynamic Self-Verify Decoding”) se plantea como un marco de decodificación con **detección paralela de alucinación** durante generación y un mecanismo de **rollback**/corrección dinámica, reportando mejoras consistentes en métricas de fidelidad/factualidad en diversos modelos y evaluaciones. ¹⁹

Un cuarto componente opcional (útil como “bajo-costo” en riesgo medio) es incorporar técnicas de **decodificación orientada a factualidad** sin retrieval, como **DoLa** (contrastar logits de capas para “hacer emergir” conocimiento factual), que reporta mejoras en veracidad en benchmarks como TruthfulQA. ²⁰

Finalmente, tu condición de riesgo alto (“obliga grounding o devuelve ‘no sé’ operativo + plan”) está alineada con el giro reciente hacia **abstención** y “enseñar al modelo a saber cuándo no sabe”. Hay estudios y surveys dedicados a abstención en LLMs y a la evaluación de esa capacidad en lugar de medir sólo exactitud. ¹³

Banco de pruebas, métricas y ablations

Tu lista de deliverables experimentales es correcta; lo que la vuelve “investigación fuerte” es **amarrarla a benchmarks existentes** (para comparabilidad) y definir un protocolo que separe: (i) errores inevitables por falta de evidencia, (ii) errores por mala calibración (sobreconfianza) y (iii) errores por drift inter-módulo.

Componentes recomendados para tu “bench propio” (combinable por suites)

Para no reinventar todo, una suite razonable puede triangular:

- **Verdad vs sesgos de entrenamiento (truthfulness):** TruthfulQA fue diseñado para medir si el modelo evita respuestas falsas que imitan falsas creencias humanas; incluye 817 preguntas en 38 categorías (salud, ley, finanzas, política, etc.), lo cual es ideal para tu eje “alto riesgo → abstenerse/grounding”. ²¹
- **Hallucination eval a gran escala:** HalluEval propone un benchmark con muestras generadas y anotadas para evaluar reconocimiento/detección de alucinación. ²²
- **Factualidad granular (soporte por evidencia):** FActScore mide porcentaje de “hechos atómicos” apoyados por fuentes; es casi exactamente tu “tasa de afirmaciones sin soporte cuando hay contexto disponible”. ¹²
- **QA con estructuras propensas a inventar listas:** CoVe reporta evaluación en preguntas tipo lista generadas desde Wikidata y en closed-book MultiSpanQA, lo que te sirve para reproducir casos donde el modelo “rellena” entidades plausibles. ²³
- **Código (verificación por ejecución):** HumanEval se introdujo para medir corrección funcional de programas generados desde docstrings y se volvió estándar para evaluación ejecutable; es ideal para tu componente “alucinación en código” porque puedes medir con tests. ²⁴
- **Código básico y cobertura amplia:** MBPP (Mostly Basic Programming Problems) se introdujo junto con análisis de escalado en síntesis de programas; su definición (974 tareas) lo hace adecuado para medir fallos “pequeños pero reales” donde el modelo inventa detalles de API o lógica. ²⁵

Métricas alineadas con tus objetivos

Tu set ya es bueno; para hacerlo “medible” y comparable con literatura:

- **Tasa de afirmaciones sin soporte:** implementable al estilo FActScore (atomic facts + verificación con retrieval). ¹²
- **Contradicciones inter-turno y con estado:** operacionalizable como *commitment tracking* + verificador (NLI/LLM-judge) sobre compromisos recientes; esto se justifica por taxonomías de “faithfulness hallucination”. ³
- **Calibración de riesgo:** si `hallucination_risk` gobierna abstención/escalamiento, evalúalo con métricas de selección (riesgo-cobertura). El foco en abstención como objeto evaluable está sintetizado en surveys recientes. ²⁶
- **Eficiencia condicionada:** mide costo medio (tokens, llamadas a retrieval, verificadores) vs mejoras; esto conecta con ideas de cómputo adaptativo/condicional. ¹⁶

Ablations (tu diseño, reforzado con factores de control)

Tu ablation “sin RAS vs RAS-sinthome vs RAS+verificación vs RAS+verificación+borromeo-constraint” es exactamente la estructura correcta para aislar contribución. Para que sea científicamente limpio, añade dos controles:

1) **RAG fijo vs RAG condicionado por riesgo:** para demostrar que el mérito está en el *gating*, no sólo en “usar RAG”. (RAG como técnica está bien establecida; lo nuevo es decidir *cuándo* aplicarla). ¹¹

2) **Decoding fijo vs decoding condicionado por riesgo:** por ejemplo, “greedy/beam” vs DoLa/DSVD condicionado; así separas “mejor decodificación” de “mejor política”. ²⁷

Modelo Möbius-IO y construcción de grafo con pliegues

Tu subprograma Möbius-IO se puede formalizar de manera muy directa porque existe una parametrización estándar de la banda de Möbius en \mathbb{R}^3 . Por ejemplo, MathWorld da una representación paramétrica con variables (s, t) donde $t \in [0, 2\pi]$ recorre la “vuelta” y $s \in [-w, w]$ da el ancho; en esa parametrización aparecen los términos $\cos(t/2)$ y $\sin(t/2)$ que codifican el medio giro.

²⁸ Además, la banda de Möbius es un ejemplo canónico de **superficie no orientable** (no hay definición global consistente de “lado”), lo cual es exactamente la intuición que quieras explotar para “input y output como la misma cara bajo otra vuelta”. ²⁹

De la superficie al grafo (tu idea de “papel arrugado”, en términos computacionales)

Tu regla de conectividad “geodésica + proximidad en \mathbb{R}^3 ” tiene una lectura estándar en geometría computacional y aprendizaje de variedades:

- La vecindad por **distanzia geodésica** captura conectividad “intrínseca” sobre la superficie. En malla triangular, hay algoritmos clásicos para aproximar distancias geodésicas y caminos mínimos, como el **Fast Marching Method** extendido a dominios triangulados (Kimmel & Sethian). ³⁰
- La vecindad por **proximidad euclídea en el embedding** introduce “contactos” por plegado: regiones lejanas en coordenadas (u, v) pueden volverse cercanas en \mathbb{R}^3 . En tu teoría, esto implementa shortcuts inter-zona sin “romper” el hecho de que todo vive en una sola banda abstracta. ²⁹

En términos prácticos, esto sugiere un pipeline robusto:

- 1) Muestrear nodos en el dominio $(u, v) / (s, t)$ de la banda. ³¹
- 2) Construir una malla (triangulación) y obtener embedding $F(S) \subset \mathbb{R}^3$ usando la parametrización (y luego tus “pliegues” como perturbaciones controladas). ²⁸
- 3) Calcular vecindades intrínsecas (kNN geodésico o radio geodésico) mediante fast marching o aproximaciones en mallas. ³²
- 4) Agregar aristas por proximidad euclídea en \mathbb{R}^3 (fold-edges), con salvaguardas para que no destruyan la geometría (por ejemplo, límites de densidad o penalización por “atajos demasiado agresivos” en zonas de alta curvatura/distorsión). La motivación de controlar vecindades y outliers aparece también en trabajos sobre vecindarios adaptativos en aprendizaje de variedades. ³³

Qué medir cuando comparas Möbius-prior vs no-prior

Tu hipótesis (“shortcuts → menos drift → menos contradicción/alucinación por desalineación inter-módulo”) es testeable si instrumentas: (i) métricas de coherencia/consistencia inter-turn, (ii) métricas de grounding (claim support), y (iii) métricas de costo (latencia/pasos). La parte de métricas de alucinación ya queda anclada por taxonomías y benchmarks; lo novedoso será aislar el aporte estructural del grafo. ³⁴

Borromeo-constraint y hoja de ruta de implementación

Qué es “borromeo” en términos matemáticos (para evitar ambigüedad)

El ejemplo paradigmático de un enlace Brunnian es el de los **Borromean rings**: ningún par está enlazado “directamente”, pero los tres juntos sí forman un enlace no trivial; cortar/remover uno libera a

los otros. ⁴ Ese rasgo suele describirse como “dependencia global triádica”: la estructura no se reduce a enlaces por pares. ³⁵ En invariantes, un hecho clave (muy citado en teoría de enlaces) es que los Borromean rings son *algebraically split*: sus **linking numbers por pares son cero**, mientras que la “interacción triple” se detecta con invariantes de orden superior como el **Milnor triple linking number** (o formulaciones equivalentes). ³⁶

Qué significa eso para tu constraint (sin forzar analogías indebidas)

Tu mejor lectura ingenieril es: el “borromeo-constraint” debe ser un **test de integridad** que falle cuando el sistema se vuelve “explicable sólo por pares” o cuando se rompe la coordinación triádica (RSI o sus análogos computacionales). La parte lacaniana te da la motivación del “cuarto anillo”: cuando el amarre triádico falla, el sinthome entra como operador de anudamiento. ³⁷

Tres maneras de implementar un borromeo-constraint computable (alineadas con tus opciones)

Aquí no hay una única “verdad”; la clave es que cada opción produzca un escalar “integridad” que puedas inyectar en `hallucination_risk`.

1) Borromeo como propiedad de rutas (sobre el grafo Möbius)

Define tres familias de rutas/ciclos (análogas a $\gamma_1, \gamma_2, \gamma_3$) y mide si tu sistema mantiene una dependencia que “colapsa” si desactivas cualquiera de las tres. La inspiración formal proviene del hecho de que en enlaces Brunnian la remoción de un componente trivializa el resto. ³⁵

En práctica: suelta una política “lesion study”: apaga un canal/campo (R/S/I) y observa si (a) cae coherencia global, (b) crece contradicción, (c) crece grounding_gap.

2) Borromeo como acoplamiento triádico de campos (RSI) sobre una superficie

Representa RSI como tres campos latentes que se co-regulan; tu constraint busca evitar que el sistema “dependa de dos” ignorando el tercero. La justificación topológica es que el fenómeno Borromean no se explica por enlaces de pares (linking number por pares cero), sino por un invariante de orden superior (triple). ³⁶

En práctica (proxy): puedes medir si existe “sinergia triádica” (interacción de tercer orden) en señales internas de consenso/consistencia, y penalizar soluciones donde el desempeño se sostenga por pares pero colapse al remover el tercero.

3) Möbius-tríptico con restricciones tipo Brunnian

Tu opción más “topológica literal” (tres bandas acopladas) tiene sentido si quieres que el grafo mismo tenga una propiedad de “remoción de un subgrafo trivializa conectividad global”. Esto se alinea directamente con la definición de Brunnian link (sublinks triviales). ³⁵

En práctica: es más caro, pero te permite instrumentar invariantes discretos de conectividad/sobrevivencia de ciclos al remover componentes.

Hoja de ruta técnica (amarrando Eje C + Eje B en un vNext implementable)

Sin entrar a detalles de lenguaje/framework, una ruta sólida (y testeable con ablations) se ve así:

- **Módulo de señales:** estimadores para incertidumbre (entropía + disagreement tipo SelfCheckGPT/self-consistency), grounding_gap (atomic facts al estilo FActScore cuando haya evidencia), y drift (contradicción con compromisos). ³⁸
- **Módulo RAS (policy):** router de presupuesto computacional y de verificación, inspirado en cómputo condicional y adaptativo (MoE/ACT): a bajo riesgo, ruta shallow; a medio, activar CoVe/DoLa/PRM-like scoring; a alto, forzar RAG o abstención “no sé + plan”. ³⁹
- **Módulo Möbius-IO:** generador de malla/embedding (parametrización), geodesic kNN y fold-edges por proximidad euclídea; geodesia con fast marching en mallas trianguladas. ⁴⁰

- **Módulo borromeo-constraint:** primero como “test” (métrica), luego como regularizador/ pérdida si decides entrenar el router. Sus fundamentos topológicos quedan claros: pares “no enlazados” pero triple “no trivial”. ⁴¹
- **Suite de evaluación:** TruthfulQA + HaluEval + tareas de grounding (FActScore-like) + QA estilo CoVe + código (HumanEval/MBPP). ⁴²

El resultado es coherente con tu objetivo: no “eliminar” toda alucinación como ideal abstracto, sino volverla **operacionalmente controlable** mediante (i) señales, (ii) gating, (iii) verificación escalada y (iv) un prior estructural (Möbius + pliegues) cuyo efecto se mide y se ablatea. ⁴³

¹ ⁶ ³⁷ The Seminar of Jacques Lacan XXIII: Le Sinthome

https://www.lacanononline.com/wp-content/uploads/2014/11/Seminar-XXIII-The-Sinthome-Jacques-Lacan-Thurston-translation.pdf?utm_source=chatgpt.com

² JOYCE & THE SINTHOME: AIMING AT THE FOURTH ...

https://londonssociety-nls.org.uk/wp-content/uploads/joyce-sinthome.pdf?utm_source=chatgpt.com

³ ³⁴ ⁴³ A Survey on Hallucination in Large Language Models

https://arxiv.org/pdf/2311.05232?utm_source=chatgpt.com

⁴ ³⁵ Brunnian Link -- from Wolfram MathWorld

https://mathworld.wolfram.com/BrunnianLink.html?utm_source=chatgpt.com

⁵ Möbius Strip -- from Wolfram MathWorld

https://mathworld.wolfram.com/MoebiusStrip.html?utm_source=chatgpt.com

⁷ ¹⁰ On Calibration of Modern Neural Networks

https://arxiv.org/abs/1706.04599?utm_source=chatgpt.com

⁸ Self-Consistency Improves Chain of Thought Reasoning in Language Models

https://arxiv.org/abs/2203.11171?utm_source=chatgpt.com

⁹ ³⁸ SelfCheckGPT: Zero-Resource Black-Box Hallucination ...

https://aclanthology.org/2023.emnlp-main.557/?utm_source=chatgpt.com

¹¹ Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks

https://arxiv.org/abs/2005.11401?utm_source=chatgpt.com

¹² FActScore: Fine-grained Atomic Evaluation of Factual ...

https://aclanthology.org/2023.emnlp-main.741?utm_source=chatgpt.com

¹³ ²⁶ A Survey of Abstention in Large Language Models

https://direct.mit.edu/tacl/article/doi/10.1162/tacl_a_00754/131566/Know-Your-Limits-A-Survey-of-Abstention-in-Large?utm_source=chatgpt.com

¹⁴ ³⁹ Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer

https://arxiv.org/abs/1701.06538?utm_source=chatgpt.com

¹⁵ ¹⁶ Adaptive Computation Time for Recurrent Neural Networks

https://arxiv.org/abs/1603.08983?utm_source=chatgpt.com

¹⁷ Chain-of-Verification Reduces Hallucination in Large Language Models

https://arxiv.org/abs/2309.11495?utm_source=chatgpt.com

¹⁸ The Lessons of Developing Process Reward Models in Mathematical Reasoning

https://arxiv.org/abs/2501.07301?utm_source=chatgpt.com

¹⁹ DSVD: Dynamic Self-Verify Decoding for Faithful ...

https://arxiv.org/abs/2503.03149?utm_source=chatgpt.com

- 20 DoLa: Decoding by Contrasting Layers Improves Factuality in Large Language Models
https://arxiv.org/abs/2309.03883?utm_source=chatgpt.com
- 21 42 TruthfulQA: Measuring How Models Mimic Human Falsehoods
https://arxiv.org/abs/2109.07958?utm_source=chatgpt.com
- 22 HaluEval: A Large-Scale Hallucination Evaluation ...
https://arxiv.org/abs/2305.11747?utm_source=chatgpt.com
- 23 Chain-of-Verification Reduces Hallucination in Large ...
https://www.research-collection.ethz.ch/server/api/core/bitstreams/468e77de-b21f-4ede-b179-8a52b01a1c5a/content?utm_source=chatgpt.com
- 24 Evaluating Large Language Models Trained on Code
https://arxiv.org/abs/2107.03374?utm_source=chatgpt.com
- 25 Program Synthesis with Large Language Models
https://arxiv.org/abs/2108.07732?utm_source=chatgpt.com
- 27 DOLA: DECODING BY CONTRASTING LAYERS ...
https://proceedings.iclr.cc/paper_files/paper/2024/file/edc36117f795ca52a0cbf6a7b3882859-Paper-Conference.pdf?utm_source=chatgpt.com
- 28 31 40 Möbius Strip -- from Wolfram MathWorld
<https://mathworld.wolfram.com/MoebiusStrip.html>
- 29 Möbius strip
https://en.wikipedia.org/wiki/M%C3%B6bius_strip?utm_source=chatgpt.com
- 30 32 R. Kimmel and J. A. Sethian Source
https://www.cis.upenn.edu/~cis6100/Kimmel-Sethian-geodesics-98.pdf?utm_source=chatgpt.com
- 33 IAN: Iterated Adaptive Neighborhoods for Manifold Learning ...
https://pmc.ncbi.nlm.nih.gov/articles/PMC12349730/?utm_source=chatgpt.com
- 36 41 Triple linking numbers, ambiguous Hopf invariants
https://vela-vick.com/wp-content/uploads/2015/05/MilnorAnn.pdf?utm_source=chatgpt.com