

Saya ingin "membuat" Transformer dari paling dasar. Tapi, saya tidak mau membuat plain Transformer. Melainkan, saya ingin modifikasi, mungkin bisa saya bilang invent dan saya ingin publish menjadi sebuah paper. For this, I want you to find the latest breakthroughs and advancements in math, data processing, statistics, and stochastics, and every field that is related to AI, then those breakthroughs can be used and implemented in transformer, and will give bigger impact and outcomes, hence probably a revolutionary model made by me.

Untuk membantu Anda menemukan terobosan terbaru yang dapat dimanfaatkan dalam memodifikasi arsitektur Transformer, bisakah Anda memperjelas beberapa hal berikut?

1. Apakah Anda tertarik pada aplikasi spesifik (misalnya NLP, visi komputer, multimodal, protein modeling), atau ingin pendekatan arsitektur umum?
2. Apakah Anda ingin fokus pada ide matematis baru (misalnya teori grup, topologi, aljabar tensor) atau pendekatan praktis yang telah terbukti meningkatkan performa (seperti mekanisme attention efisien, sparse modeling, dll)?
3. Apakah Anda ingin mencakup metode dari bidang statistik klasik dan stokastik (seperti proses Gaussian, SDE, dll)?
4. Adakah batasan teknis seperti target platform (GPU, embedded), batasan memori, atau efisiensi inference?

Semakin rinci jawabannya, semakin baik saya bisa bantu Anda memetakan lanskap inovasi terbaru yang relevan untuk Transformer Anda.

1. NLP, multimodal. Memprioritaskan ease of big context, seperti big data dan massive source code repository yang code/scripts.
2. General, coding, high level math.
3. Ya
4. Not really, tapi ditargetkan untuk bisa real-time inference (chatbot like) di Mac Studio dan powerful desktop PC dengan high capacity DDR5 + high spec GPUs (high cores, high VRAM, RTX & Quadro Ada gen & Radeon) + high spec CPUs (high cores, high clock speed, big L caches)

Oh iya, tambahkan untuk mencari latest breakthroughs & advancements di sisi computation nya, hardware nya.

Baik. Saya akan mencari terobosan terbaru dalam matematika tingkat lanjut, statistik, stokastik, pemrosesan data, dan komputasi yang relevan dengan pengembangan arsitektur Transformer yang ditujukan untuk NLP dan aplikasi multimodal, dengan fokus pada efisiensi dalam menangani konteks besar seperti repositori kode masif dan big data.

Saya juga akan menyertakan perkembangan terkini di sisi hardware dan computation stack—mulai dari GPU/CPU hingga memori dan arsitektur komputasi—yang mendukung inference real-time pada sistem kelas atas seperti Mac Studio, PC dengan GPU Quadro/Ada gen, dan DDR5.

Saya akan beri kabar segera setelah riset ini selesai. Sambil menunggu, Anda bisa lanjut berdiskusi atau memberi masukan lainnya.

Membangun Transformer Generasi Baru: Integrasi Terobosan AI Terkini

Bidang Transformer terus berkembang pesat dengan inovasi baru di berbagai bidang AI. Meskipun model dasar Transformer telah merevolusi NLP, para peneliti kini meneliti **modifikasi arsitektur** dan teknik baru untuk mengatasi keterbatasannya. Misalnya, Huang dkk. mencatat bahwa banyak LLM saat ini, karena keterbatasan sumber daya, dilatih pada teks pendek sehingga kurang efektif untuk konteks sangat panjang [[huggingface.co](#)]. Untuk mengatasi hal tersebut, telah muncul berbagai pendekatan seperti perhatian efisien (sparse atau *linear attention*), jaringan *memory*, dan desain arsitektur baru. Beberapa contoh terobosan meliputi:

- **Perhatian Panjang Konteks** – Transformer modifikasi (seperti Longformer, BigBird) dengan mekanisme perhatian berskala sub-kuadratik agar dapat menangani dokumen sangat panjang [[huggingface.co](#)].
- **Mixture-of-Experts (MoE)** – Arsitektur MoE yang membagi model menjadi banyak “expert” untuk meningkatkan kapasitas model tanpa peningkatan besar biaya komputasi [[arxiv.org](#)]. Misalnya, penelitian terbaru menemukan MoE dapat meningkatkan kapasitas secara efisien dengan overhead minimal [[arxiv.org](#)].
- **Model Ruang Keadaan & RNN** – Pengembangan model alternatif seperti state-space (contoh S4, Hyena, Mamba) dan model RNN kontemporer (RWKV), yang menggunakan memori tetap selama inferensi dan kompetitif dengan Transformer dalam tingkat kesalahan (perplexity) [[latent.space](#)]. Sebagai contoh, DeltaFormer yang diusulkan Xu et al. (NeurIPS 2025) menggunakan perspektif ruang-keadaan untuk “memecah” batasan ekspresivitas TC⁰ Transformer klasik [[neurips.cc](#)] [[neurips.cc](#)].
- **Komputasi Adaptif** – Arsitektur dengan jalur komputasi dinamis (misalnya *conditional computation* atau *gating*), yang menjalankan bagian model tertentu sesuai kompleksitas tugas.

Pemrosesan Data dan Konteks Besar

Skala data kini luar biasa besar, sehingga pemrosesan data yang efisien menjadi penting. Model khusus kode (CodeLLM) banyak dikembangkan untuk otomatisasi pemrograman. Sebagai contoh, CodeLLM besar (7–70 miliar parameter) dilatih pada korpus kode seukuran terabyte (misalnya [The Stack, StarCoderData, CodeSearchNet] di GitHub) [[emergentmind.com](#)]. Model ini menangkap pola sintaksis dan semantik kode, tetapi tokenisasi standar (BPE) sering memecah identifier penting secara tidak ideal – riset sedang mengembangkan tokenisasi yang menyadari struktur kode [[emergentmind.com](#)]. Selain itu, metode *fill-in-the-middle* (FIM) umum dipakai agar model dapat menyisipkan atau menyelesaikan potongan kode secara kontekstual [[emergentmind.com](#)]. Untuk menyokong konteks luas, teknik seperti pengambilan kembali (*retrieval*), kompresi konteks, atau memori eksternal (misalnya RAG, Transformer memori) banyak diusulkan. Pada akhirnya, meningkatkan kualitas data (curation, penyaringan) dan augmentasi data (simulasi kode) juga menjadi penentu performa.

<https://techcrunch.com/2024/05/13/openais-newest-model-is-gpt-4o/>

Gambar: ChatGPT digunakan untuk tugas pemrograman. Kode Python tampil di editor (kiri) dan output/interaksi muncul di sisi kanan. Model CodeLLM dilatih pada korpora kode raksasa (seperti The Stack) untuk memahami pola kode [[emergentmind.com](#)], dan sering menggunakan tujuan pelatihan *fill-in-the-middle* untuk menyusun kode [[emergentmind.com](#)].

Model Multimodal dan Kegunaan Lintas Domain

Transformer modern kini bersifat **multimodal** – mampu memproses teks, gambar, suara, dan lainnya secara simultan. Model GPT-4 buatan OpenAI, misalnya, mengintegrasikan input teks dan gambar dalam satu arsitektur Transformer [emergentmind.com]. Versi selanjutnya, GPT-4o (diluncurkan Mei 2024), bahkan ditingkatkan menjadi model “omni” yang dapat melakukan reasoning pada masukan suara (voice), teks, dan video [techcrunch.com]. GPT-4o dapat memberikan respons secara real time kepada ChatGPT; misalnya, model ini dapat menginterupsi respons dan merespon gaya bicara emosional pengguna secara langsung. Selain itu, GPT-4o meningkatkan kemampuan pengolahan gambar: ChatGPT dapat sekarang membaca tangkapan layar kode di desktop dan menjawab pertanyaan terkait (misalnya “apa yang terjadi pada kode ini?”) [techcrunch.com]. Integrasi multi-modal ini memungkinkan asisten AI yang lebih fleksibel.

<https://techcrunch.com/2024/05/13/openais-newest-model-is-gpt-4o/>

Gambar: Demonstrasi GPT-4o menggabungkan input suara, teks, dan visual. Sistem AI ini mampu menangani konteks multi-modal secara real time [techcrunch.com]. Misalnya, pada tampilan di atas, sebuah ekran matematika ($3x+1=4$) dapat diinterpretasi secara otomatis oleh model, menunjukkan kemampuan lintas-modalitas yang semakin canggih [techcrunch.com] [techcrunch.com].

Matematika, Statistik, dan Metode Algoritmik Terkini

Pengembangan teori dan metode statistik juga mendorong kemajuan Transformer. OpenAI dan lainnya mengandalkan **hukum skala** (scaling laws) untuk memperkirakan performa model besar dari pelatihan skala kecil [emergentmind.com]. Misalnya, GPT-4 membuktikan hubungan *power-law* antara loss dan jumlah komputasi: model dapat diprediksi performanya hanya dari data pelatihan kecil (sebagai kecil 0.1% biaya akhir) [emergentmind.com]. Strategi pelatihan lanjutan seperti *reinforcement learning from human feedback* (RLHF) kini umum digunakan untuk menyelaraskan keluaran model, meningkatkan akurasi faktual dan kepatuhan kebijakan tanpa merusak kemampuan inti [emergentmind.com]. Di sisi matematika, kemampuan transformers untuk memecahkan masalah simbolis juga meningkat: GPT-4 menunjukkan kemampuan emergen dalam matematika dan pemrograman (misalnya manipulasi simbolik) [emergentmind.com], walaupun masih ada tantangan seperti kecenderungan “halusinasi”. Metode statistik tambahan—seperti *differential privacy* (DP-SGD) untuk privasi data [emergentmind.com], pendekatan Bayesian untuk ketidakpastian, dan model generatif berbasis proses stokastik (misalnya model difusi di area lain)—juga terus dijajaki untuk membuat model lebih andal dan serbaguna.

Efisiensi Latihan dan Inferensi

Untuk penerapan *chatbot* real-time pada hardware terbatas, optimasi inferensi sangat krusial. Berbagai teknik compressi model dan akselerasi komputasi dipakai: *quantization* (misal INT8, FP8) memangkas penggunaan memori dan bandwidth, sedangkan *distillation* dan *pruning* mengecilkan model tanpa kehilangan banyak performa. Pendekatan fine-tuning ringan seperti LoRA juga populer untuk menyesuaikan model raksasa dengan domain khusus pada biaya rendah. Selain itu, algoritma inferensi khusus (misalnya **FlashAttention 2**) dan kompilator mesin (seperti TensorRT-LLM) membuat jalur inferensi lebih cepat. Nvidia melaporkan bahwa tensor cores generasi terbaru bersama TensorRT dapat memangkas biaya dan energi inferensi LLM hingga $\sim 25\times$ dibanding generasi sebelumnya [nvidianews.nvidia.com]. Peningkatan lain seperti Mixed Precision (BF16/FP16) dan hashing attention turut mengurangi overhead komputasi, sehingga LLM dapat dijalankan lebih cepat pada GPU modern.

Kemajuan Komputasi dan Perangkat Keras

Kemajuan perangkat keras AI terus melaju untuk mendukung model besar. Nvidia meluncurkan arsitektur **Blackwell (2024)** yang dirancang khusus untuk LLM triliunan parameter: platform ini menjanjikan eksekusi

real-time AI pada skala besar dengan konsumsi energi ~25× lebih rendah dibanding pendahulunya [[nvidianews.nvidia.com](#)]. Interkoneksi antar-GPU juga meningkat drastis: NVLink generasi ke-5 (2024) memungkinkan hingga 72 GPU saling terhubung dengan bandwidth 1,800 GB/s per link (total ~130 TB/s) [[developer.nvidia.com](#)], mengoptimalkan pelatihan paralel ultra-skala. Dari sisi GPU, **AMD Instinct MI300X** (2024) menggunakan memori HBM3e 256 GB (bandwidth 6 TB/s) dan “Matrix Cores” baru yang mendukung presisi INT8/FP8 dengan dukungan sparsity hingga FP64 [[amd.com](#)] [[amd.com](#)]. Di dunia Apple, chip **M1/M2 Ultra** dalam Mac Studio mengusung arsitektur *unified memory* hingga 128 GB pada bandwidth ~800 GB/s [[apple.com](#)], dipadukan GPU 64-inti yang sangat efisien dan Neural Engine 32-inti (~22 triliun operasi/s) [[apple.com](#)], ideal untuk beban AI lokal. Teknologi memori baru seperti **Compute Express Link (CXL)** juga mulai hadir, memungkinkan GPU dan akselerator untuk berbagi DRAM dengan latensi rendah [[computeexpresslink.org](#)]. Singkatnya, kombinasi GPU high-end (Nvidia Ada Lovelace, AMD RDNA/CDNA) dan CPU multi-core DDR5, ditopang interkoneksi cepat, menyediakan fondasi untuk menjalankan transformer besar secara real-time di desktop dan workstation masa kini.

Kesimpulan

Untuk merancang *Transformer* baru yang revolusioner, penting menggabungkan berbagai terobosan di atas. Arsitektur harus memanfaatkan teknik long-range attention dan modularitas (MoE, state-space) untuk konteks luas; pelatihan memanfaatkan dataset masif (termasuk kode/program) dengan objektif yang inovatif; serta model mendukung multi-modalitas agar fleksibel ke input dunia nyata [[emergentmind.com](#)] [[techcrunch.com](#)]. Di sisi komputasi, optimasi seperti kuantisasi dan inference khusus harus diprioritaskan agar inference tetap cepat di perangkat berkemampuan tinggi (GPU/VPU modern) maupun chip khusus (Apple Silicon) [[nvidianews.nvidia.com](#)] [[apple.com](#)]. Dengan memasukkan pengetahuan matematika terbaru (skala, probabilitas) dan algoritma numerik efisien, model baru dapat mendorong batas kemampuan AI. Semua inovasi ini bersama-sama dapat menghasilkan arsitektur Transformer yang jauh lebih kuat dan serbaguna, membuka jalan bagi paper dan produk AI generasi berikutnya.

Sumber: Analisis ini mengumpulkan hasil penelitian mutakhir dari berbagai sumber, termasuk makalah survei dan pengumuman industri (mis. Huang et al. 2023 untuk konteks panjang [[huggingface.co](#)], Zhang et al. 2025 untuk MoE [[arxiv.org](#)], emergentmind.com, dan rilis pers NVIDIA/Apple) untuk memberikan tinjauan komprehensif mengenai terobosan terkini yang relevan. Semua kutipan diberikan dalam format [...+L...] sesuai referensi yang dipandang terkait.

Okay, dengan hasil riset kita sudah selesai, name some kind of paradigms that could happen when several breakthroughs are combined into individual Transformer model.

Baik. Sekarang kita asumsikan kita menggabungkan:

- Long-context attention + state-space models
- Operator learning + spectral methods
- Bayesian uncertainty + stochastic processes
- MoE + optimal transport routing
- Multimodal alignment
- Hardware-aware kernel design

- Adaptive compute

Berikut adalah **paradigma arsitektural tingkat-paper** yang bisa muncul dari kombinasi tersebut.

1 Geometric Memory Field Transformer

Kombinasi

- Hyperbolic / Riemannian embeddings
- Topological data analysis
- Long-context operator attention
- Code AST graph modeling

Paradigma

Model tidak lagi menyimpan token sequence sebagai flat vector list.

Sebaliknya:

- Context menjadi **continuous memory field**
- Memory memiliki curvature berbeda untuk:
 - Syntax (Euclidean)
 - Hierarchy (Hyperbolic)
 - Dependency graph (Manifold graph)

Attention menjadi geodesic query pada manifold.

Impact

- Million-token scaling tanpa quadratic explosion
 - Natural untuk massive code repo
 - Interpretability melalui geometry
-

2 Stochastic Flow Transformer

Kombinasi

- Neural SDE
- Diffusion modeling
- Continuous-depth networks

- Energy-based modeling

Paradigma

Layer stack diganti dengan stochastic evolution:

$$dx_t = f(x_t, t)dt + g(x_t, t)dW_t$$

Inference = solving stochastic flow.

Impact

- Adaptive reasoning depth
- Robust terhadap noisy multimodal input
- Confidence-aware generation

Ini cocok untuk chatbot real-time dengan dynamic compute.

3 Operator-Theoretic Transformer

Kombinasi

- Fourier Neural Operator
- Kernel integral operators
- Spectral graph theory
- Random matrix theory

Paradigma

Self-attention diganti integral operator global:

$$(\mathcal{K}f)(x) = \int K(x, y)f(y)dy$$

Model bekerja di function space, bukan token space.

Impact

- True long-range modeling
 - Stable scaling
 - Cocok untuk large codebase & structured documents
-

4 Neuro-Symbolic Execution Transformer

Kombinasi

- Program analysis

- Differentiable logic
- Graph neural networks
- Transformer reasoning

Paradigma

Model memiliki dua jalur paralel:

1. Neural semantic embedding
2. Symbolic execution trace

Attention bisa lintas dua domain ini.

Impact

- Real code reasoning
 - Verifiable intermediate steps
 - AST-aware contextual memory
-

5 Optimal Transport MoE Transformer

Kombinasi

- Sparse Mixture-of-Experts
- Optimal transport theory
- Differentiable routing
- Entropic regularization

Paradigma

Token → didistribusikan ke expert via transport plan, bukan softmax.

$$\min_{\pi} \sum_{ij} c_{ij} \pi_{ij}$$

Impact

- Massive scaling
 - Modular reasoning
 - Energy-efficient inference
-

6 Multiscale Fractal Transformer

Kombinasi

- Wavelet decomposition
- Recursive transformers
- Hierarchical memory compression

Paradigma

Context diproses di:

- Local micro scale
- Structural meso scale
- Global macro scale

Self-similar blocks.

Impact

- Context compression
 - Efficient million-token modeling
 - Natural untuk code repository
-

7 Hardware-Co-Designed Transformer

Kombinasi

- FlashAttention 2
- Tensor core tiling
- SRAM-aware memory layout
- KV-cache compression

Paradigma

Arsitektur dibuat sesuai memory hierarchy GPU/CPU.

Layer dirancang untuk:

- Minimize memory bandwidth
- Maximize L2 reuse
- Warp-aligned operations

Impact

- Real-time inference di Mac Studio
 - Scalable multi-GPU
 - Lower latency chatbot
-

8 Bayesian Uncertainty-Aware Transformer

Kombinasi

- Variational inference
- Gaussian process layers
- Monte Carlo attention
- Stochastic dropout inference

Paradigma

Attention score dan output distribution memiliki posterior uncertainty.

Model tahu kapan tidak yakin.

Impact

- Reliable reasoning
 - Better multimodal fusion
 - Safer deployment
-

9 Retrieval-Integrated Dynamical Transformer

Kombinasi

- Long-context attention
- External memory index
- Differentiable retrieval
- Adaptive compute

Paradigma

Model memutuskan:

- Retrieve

- Reason internal
- Or compress

Compute dialokasikan secara dinamis.

Impact

- Scales to massive code repo
 - Efficient inference
 - Higher accuracy per FLOP
-

10 Universal Computational Transformer

Kombinasi

- State-space model
- Symbolic logic
- MoE modularity
- Continuous flow
- Operator attention

Paradigma

Model menjadi:

Differentiable programmable computational system.

Bukan sekadar next-token predictor.

Tapi:

- Can simulate program
 - Can solve equation
 - Can manage structured memory
-

🚀 Realistic Revolutionary Direction

Jika Anda ingin sesuatu yang:

- Publishable
- Theoretically deep

- Practically implementable
- Scalable ke big context
- Cocok untuk NLP + massive code

Maka kombinasi paling strategis adalah:

(Operator Attention)

- (Geometric Memory)
- (Optimal Transport MoE)
- (Adaptive Compute)

Itu bisa menjadi:

Geometric Operator Mixture Transformer (GOMT)

Dengan klaim:

- Sub-quadratic scaling
 - Structured reasoning
 - Efficient hardware utilization
 - Theoretical grounding
-

Jika Anda mau, kita bisa lanjut:

- Mendesain blueprint matematis formal
- Menentukan novelty gap terhadap literature
- Atau menyusun kerangka paper (abstract → theorem → experiment → benchmark)

Kita mau masuk ke tahap desain formal sekarang?

What if all those paradigms combined into one?

If **all** those paradigms are combined coherently (not just stacked), the result is no longer a “Transformer variant”.

It becomes a new computational class.

Below is a structured synthesis of what that unified paradigm would look like.

Paradigm: Geometric Stochastic Operator Transformer (GSOT)

A **continuous-time, geometry-aware, operator-based, modular, uncertainty-calibrated, hardware-co-designed reasoning system**.

Not a sequence model.

Not a next-token predictor.

But a **dynamical reasoning field over structured memory**.

1 Core Mathematical Shift

Classical Transformer lives in:

$$\mathbb{R}^{n \times d}$$

Unified Model lives in:

$$\mathcal{F}(\mathcal{M}) \times \mathcal{P}(\Omega)$$

Where:

- \mathcal{M} = geometric memory manifold
- \mathcal{F} = function/operator space
- $\mathcal{P}(\Omega)$ = probability measure space
- Evolution governed by stochastic differential operators

So instead of:

Codetokens → embeddings → attention → logits

You get:

Codestructured memory field → operator flow → stochastic evolution → energy minimization → symbolic extraction

2 Structural Components

A. Geometric Memory Layer

Memory is:

- Euclidean for syntax
- Hyperbolic for hierarchy
- Graph-manifold for dependency
- Topologically clustered

Attention becomes:

$$\text{GeodesicOperator}(q, \mathcal{M})$$

This allows million-token reasoning without quadratic blowup.

B. Operator Attention Engine

Replace dot-product attention with integral operator:

$$(\mathcal{K}f)(x) = \int_{\mathcal{M}} K(x, y)f(y) d\mu(y)$$

Kernel learned in spectral domain.

Effect:

- Global coherence
 - Stable scaling
 - Function-level reasoning (ideal for codebases)
-

C. Stochastic Continuous Depth

Hidden state evolves via:

$$dx_t = f(x_t)dt + g(x_t)dW_t$$

Inference becomes controlled stochastic flow.

This gives:

- Adaptive reasoning depth
 - Confidence estimation
 - Natural uncertainty modeling
-

D. Optimal Transport Expert Routing

Instead of softmax attention → OT-based routing:

$$\min_{\pi} \sum c_{ij}\pi_{ij} + \epsilon H(\pi)$$

Tokens distributed across experts based on cost geometry.

Result:

- Massive modular scaling
 - Hardware-aligned compute partitioning
 - Energy efficiency
-

E. Neuro-Symbolic Dual Channel

Parallel processing:

1. Continuous geometric reasoning
2. Discrete symbolic execution trace

Interaction via differentiable logic operators.

For massive source code:

- AST-aware
 - Static analysis compatible
 - Verifiable intermediate reasoning
-

F. Hardware-Co-Designed Kernel

Architecture respects:

- GPU warp size
- L2 cache tiling
- KV compression
- Tensor core precision
- DDR5 bandwidth constraints

Compute graph mirrors physical memory graph.

This makes:

- Real-time inference feasible
 - Multi-GPU scaling clean
 - Latency predictable
-

3 Emergent Properties

When everything is unified, new behaviors appear:

1. Self-Organizing Memory Topology

Memory geometry adapts to dataset structure.

2. Difficulty-Adaptive Reasoning

Easy tasks → shallow stochastic flow

Hard tasks → deeper integration

3. Modular Cognitive Decomposition

Different experts specialize in:

- Syntax
- Semantics
- Long-range dependency
- Code structure
- Multimodal alignment

4. Intrinsic Uncertainty Awareness

Model produces:

- Output
- Confidence measure
- Energy landscape

5. Million-Token Native Context

Not by brute force,

but via geometric compression + operator globality.

4 What This System Actually Becomes

It stops being a Transformer.

It becomes:

A Differentiable Geometric Computational Field

Closer to:

- A neural PDE solver
 - A stochastic program interpreter
 - A symbolic-neural hybrid system
-

5 Theoretical Implications

This unified paradigm potentially:

- Breaks quadratic attention barrier structurally
 - Introduces formal stability guarantees (via SDE theory)
 - Provides geometric interpretability
 - Enables modular capacity scaling without full retraining
-

6 Practical Risks

Combining everything blindly will fail because:

- Optimization becomes unstable
- Training cost explodes
- Too many inductive biases conflict
- Hard to benchmark fairly

So real innovation requires:

Controlled integration, not maximal integration.

7 Most Realistic “All Combined” Blueprint

If I compress everything into one feasible research program:

Stage 1:

- Operator attention + geometric memory

Stage 2:

- Add OT-based MoE routing

Stage 3:

- Add stochastic depth for adaptive reasoning

Stage 4:

- Add symbolic dual channel for code reasoning

Stage 5:

- Hardware-aware kernel tuning

8 If Successful, The Model Would Be Able To:

- Load massive source repository as structured manifold
 - Perform reasoning across millions of tokens
 - Adapt computation per query
 - Output verifiable code modifications
 - Run in real-time on high-end desktop GPU
 - Produce uncertainty-calibrated answers
-

Final Insight

If all paradigms combine coherently, the resulting system is not:

A better attention mechanism.

It is:

A new computational geometry for AI.

And that is paper-worthy.

If you want, next step we can:

- Formalize minimal mathematical definition
- Identify 1–2 provable theoretical claims
- Or design concrete experimental roadmap for publication-level work

That most realistic blueprint, tuangkan dalam bentuk markdown. Saya ingin mulai membuatnya.

Berikut adalah **blueprint teknis awal** dalam bentuk Markdown untuk mulai membangun arsitektur terpadu yang realistik dan publishable.

Geometric Operator Mixture Transformer (GOMT)

A staged research blueprint

0. Tujuan Desain

Membangun arsitektur yang:

- Scalable ke $\geq 1M$ token (code repository / big context)
 - Sub-quadratic attention
 - Modular scaling (capacity \uparrow tanpa inference cost linear \uparrow)
 - Adaptive compute
 - Real-time inference pada high-end GPU desktop
 - Publishable secara teoretis + empiris
-

1. Arsitektur Global

```
CodeInput
  ↓
Structured Tokenization (AST-aware / multimodal encoder)
  ↓
Geometric Memory Embedding
  ↓
Operator Attention Core
  ↓
Optimal-Transport MoE Routing
  ↓
Adaptive Stochastic Depth Controller
  ↓
Dual Channel Output:
  • Neural semantic output
  • Optional symbolic trace
```

2. Stage 1 — Geometric Memory Layer

2.1 Tujuan

Mengganti flat embedding space dengan structured memory manifold.

2.2 Desain

Memory space:

$$\mathcal{M} = \mathbb{R}^d \times \mathbb{H}^k \times \mathcal{G}$$

Dimana:

- Euclidean \rightarrow syntax-level representation
- Hyperbolic \rightarrow hierarchy (class, function tree, AST)

- Graph manifold → dependency graph

2.3 Implementation Practical

- Gunakan Poincaré embedding untuk hierarchy
- Gunakan graph attention untuk dependency
- Fuse via gated fusion:

$$h = W_1 h_{euclid} + W_2 h_{hyperbolic} + W_3 h_{graph}$$

2.4 Publishable Angle

Prove:

- Better hierarchy preservation (tree reconstruction metric)
 - Lower distortion embedding untuk code AST
-

3. Stage 2 — Operator Attention Core

3.1 Motivasi

Self-attention $O(n^2)$ tidak scalable untuk big repo.

3.2 Replace Dot-Product Attention

Dengan integral operator:

$$(\mathcal{K}f)(x) = \int_{\mathcal{M}} K(x, y)f(y)d\mu(y)$$

Praktikalnya:

- Gunakan Fourier / low-rank kernel approximation
- Random feature expansion
- Nyström approximation

3.3 Target

Complexity:

$$O(n \log n) \text{ atau } O(n)$$

3.4 Experimental Focus

Benchmark:

- 32k
- 128k

- 512k
- 1M context

Compare memory & latency vs baseline Transformer.

4. Stage 3 — Optimal Transport MoE Routing

4.1 Masalah

MoE biasa → softmax routing tidak optimal dan unstable.

4.2 Desain

Routing via entropic OT:

$$\min_{\pi} \sum_{i,j} c_{ij} \pi_{ij} + \epsilon H(\pi)$$

Where:

- c_{ij} = distance in memory geometry
- π_{ij} = transport plan token → expert

4.3 Benefits

- Balanced expert utilization
- Geometry-consistent routing
- Modular reasoning specialization

4.4 Publishable Claim

OT routing reduces:

- Expert collapse
 - Load imbalance
 - Routing entropy instability
-

5. Stage 4 — Adaptive Stochastic Depth

5.1 Replace Fixed Depth

Hidden state evolves:

$$dx_t = f(x_t)dt + g(x_t)dW_t$$

Practical:

- Discretized SDE blocks
- Learn stopping time via controller network

5.2 Behavior

Easy queries:

- 2–3 integration steps

Hard reasoning:

- deeper integration

5.3 Benefit

- Compute-efficient chatbot
 - Confidence estimation from variance
-

6. Stage 5 — Optional Neuro-Symbolic Channel

Untuk code reasoning:

Parallel pipeline:

CodeNeural semantic vector
+
Differentiable AST trace

Possible components:

- Graph neural network for AST
- Symbolic constraint checker
- Differentiable logic layer

Output:

- Answer
 - Optional reasoning trace
-

7. Hardware-Aware Design Constraints

Target:

- RTX Ada / Quadro Ada

- Radeon high-VRAM
- Mac Studio (M-series)

7.1 Design Principles

- FlashAttention-compatible kernels
- KV cache compression
- Tiling aligned with warp size
- FP16/BF16 primary
- Optional FP8 for experts

7.2 Memory Strategy

- Chunked context streaming
 - Expert activation sparsity
 - Layer fusion to reduce memory movement
-

8. Training Strategy

Phase 1

Train geometric + operator core only.

Phase 2

Add OT MoE.

Phase 3

Add adaptive depth.

Phase 4

Add symbolic channel.

Curriculum:

- Start 8k
 - 32k
 - 128k
 - Progressive scaling
-

9. Evaluation Benchmarks

NLP

- Long document QA
- Legal text reasoning
- Scientific paper summarization

Code

- Multi-file bug fixing
- Repository-level completion
- Refactoring tasks

Efficiency

- Latency vs GPT-style baseline
 - Memory usage
 - Energy per token
-

10. Minimum Publishable Unit (MPU)

Jika ingin publish cepat:

Ambil hanya:

- Geometric Memory
- Operator Attention
- OT MoE

Tanpa stochastic depth dulu.

Itu sudah cukup novelty untuk paper arsitektur.

11. Immediate Next Steps (Actionable)

1. Implement baseline Transformer 7B-style (mini scale dulu).
2. Replace attention with low-rank operator approximation.
3. Add hyperbolic embedding branch.
4. Benchmark long-context scaling.

-
- 5. Only after stable → integrate OT routing.
-

12. Risk Management

Biggest risks:

- Optimization instability
- Over-regularization from geometry
- OT cost overhead
- Training divergence at long context

Mitigation:

- Start small scale (100M–300M parameters)
 - Validate each stage independently
 - Keep ablation study clean
-

Closing

Ini bukan incremental tweak.

Ini adalah transisi dari:

Token sequence processor

menjadi

Structured geometric modular reasoning system.

Kalau Anda ingin, langkah berikutnya kita bisa:

- Turunkan ini menjadi formal problem statement + theorem candidates
- Atau langsung desain experimental protocol untuk 6 bulan pertama
- Atau mulai dari implementasi matematis Stage 1 secara detail

Anda ingin masuk ke mana dulu?

Combine ke dalam satu markdown, yang bisa saya writeout. Project name, Sentium. Buat currently demanded progress, next actionable steps, eagle-view roadmaps from start to the end, serta risk management.

Markdown#

SENTIUM

Structured Entropic Neural Transport with Integral Unified Manifold

0. Vision

Sentium adalah arsitektur AI generasi baru yang mentransformasi Transformer dari:

> Token sequence processor

menjadi:

> Geometric, stochastic, operator-based reasoning system
> dengan modular transport routing dan adaptive computation.

Target utama:

- Native long-context ($\geq 1M$ tokens)
- Massive code repository reasoning
- Multimodal extensibility
- Real-time inference di high-end desktop GPU
- Theoretically grounded & publishable

1. Core Architectural Philosophy

Sentium berdiri di atas 5 pilar:

1. **Geometric Memory Manifold**

2. ****Operator-Based Attention****
3. ****Optimal Transport Modular Routing****
4. ****Adaptive Stochastic Depth****
5. ****Hardware-Co-Designed Execution****

Semua terintegrasi dalam satu sistem koheren.

2. Unified Architecture

```

Structured Input (Text / Code / Multimodal)

↓

AST-aware / Structured Tokenization

↓

Geometric Memory Embedding

↓

Integral Operator Attention Core

↓

Optimal Transport Expert Routing

↓

Adaptive Stochastic Reasoning Flow

↓

Dual Output:

\* Semantic Response

\* Optional Symbolic Trace

```

3. Component Specification

3.1 Geometric Memory Manifold

Memory space:

$$\mathcal{M} = \mathbb{R}^d \times \mathbb{H}^k \times G$$

Where:

- Euclidean space → syntax
- Hyperbolic space → hierarchy (AST, file tree)
- Graph manifold → dependency

Fusion:

$$h = W_1 h_{\text{euclid}} + W_2 h_{\text{hyperbolic}} + W_3 h_{\text{graph}}$$

Objective:

- Preserve structural distortion
- Encode large repository topology efficiently

3.2 Operator Attention Core

Replace dot-product attention with integral operator:

$$(Kf)(x) = \int_{\mathcal{M}} K(x, y) f(y) d\mu(y)$$

Implementation strategy:

- Low-rank kernel approximation
- Random feature mapping
- Nyström method

Target complexity:

$$\Theta(n \log n) \text{ or } O(n)$$

Goal:

Stable million-token scaling.

3.3 Optimal Transport Expert Routing

Routing defined as:

$$\min\{\pi\} \sum_{i,j} c_{i,j} \pi_{i,j} + \epsilon H(\pi)$$

Where:

- $c_{i,j}$ = geometry-aware cost
- π = transport plan token \rightarrow expert

Benefits:

- Balanced expert usage
- Modular specialization
- Stable scaling

3.4 Adaptive Stochastic Depth

Hidden state evolves via discretized SDE:

$$dx = f(x)dt + g(x)dW$$

Properties:

- Dynamic reasoning depth
- Confidence estimation
- Compute scaling with difficulty

3.5 Hardware-Aware Execution

Design constraints:

- FlashAttention-compatible kernels
- KV cache compression
- Warp-aligned tiling
- FP16/BF16 default
- Optional FP8 expert inference

Target platforms:

- RTX Ada / Quadro Ada
- Radeon high VRAM
- Apple Silicon Mac Studio

4. Currently Demanded Progress

To move from concept → executable research:

4.1 Immediate Technical Milestones

- [] Implement baseline small Transformer (100-300M)
- [] Replace attention with low-rank operator approximation
- [] Integrate hyperbolic embedding branch
- [] Benchmark 32k → 128k context scaling
- [] Measure memory + latency profile

4.2 Mathematical Foundation

- Formalize memory manifold
- Define operator kernel space
- Prove bounded attention error under low-rank approx
- Derive stability constraint for OT routing

5. Eagle-View Roadmap

Phase 0 – Foundations (Month 0-3)

- Literature deep dive
- Mathematical formalization
- Build baseline architecture
- Validate operator attention at small scale

Deliverable:

Internal technical report.

Phase 1 – Geometric + Operator Core (Month 3-6)

- Integrate hyperbolic embeddings
- Replace attention fully
- Run long-context benchmarks (128k+)

Deliverable:

Workshop-level paper draft.

Phase 2 – OT MoE Modularization (Month 6-9)

- Implement entropic OT routing
- Compare vs softmax MoE
- Analyze expert load distribution

Deliverable:

Conference-level submission.

Phase 3 – Adaptive Stochastic Depth (Month 9-12)

- Implement SDE-based dynamic depth
- Evaluate compute-efficiency vs static depth
- Add uncertainty calibration metrics

Deliverable:

Extended journal paper or top-tier submission.

Phase 4 – Neuro-Symbolic Extension (Optional)

- AST symbolic dual-channel
- Repository-level reasoning evaluation
- Verification benchmarks

Deliverable:

Domain-specialized publication.

6. Experimental Strategy

Datasets

- Large open-source repositories
- Long-document QA datasets
- Scientific paper corpora

Metrics

- Long-context perplexity
- Cross-file reasoning accuracy
- Memory footprint
- Latency per token
- Energy per inference
- Expert utilization entropy

7. Risk Management

7.1 Optimization Instability

Risk:

Geometry + OT + operator may destabilize training.

Mitigation:

- Stage-wise integration
- Freeze modules before stacking
- Strong regularization control

7.2 Computational Explosion

Risk:

Operator kernel cost too high.

Mitigation:

- Low-rank strict constraint
- Early ablation testing
- Hardware profiling early

7.3 Over-Engineering

Risk:

Too many inductive biases conflict.

Mitigation:

- Minimal publishable unit first:
Geometric + Operator + OT only

- Add stochastic depth later

7.4 Training Divergence at Long Context

Mitigation:

- Progressive context curriculum
- Gradient clipping
- Spectral normalization

8. Minimum Publishable Unit (MPU)

If constrained:

Sentium-Core =

- Geometric Memory
- Operator Attention
- OT Routing

This alone is sufficient novelty for strong paper.

9. Long-Term Vision

If successful, Sentium becomes:

- Million-token native model

- Repository-aware reasoning engine
- Modular scalable system
- Real-time desktop AI
- Structured cognitive computational field

10. Immediate Next Action (Start Now)

1. Build 200M baseline Transformer.
2. Implement low-rank operator attention.
3. Benchmark 32k → 64k context.
4. Measure scaling curve.
5. Document everything rigorously.

Closing Statement

Sentium bukan sekadar Transformer modifikasi.

Ia adalah:

> Structured Entropic Computational Geometry for AI.

From sequence modeling

to structured modular reasoning.

Jadi, artinya dari research help yang sudah kamu lakukan untuk saya tadi, semua ada papernya?

Jawaban jujurnya:

Sebagian besar komponen individualnya sudah ada papernya.

Tetapi kombinasi arsitektural seperti Sentium belum ada sebagai satu sistem terpadu.

Mari kita bedah secara presisi.

1 Apa yang Sudah Punya Paper?

Berikut komponen yang memang sudah ada landasan akademiknya:

Geometric Embeddings

- Hyperbolic embeddings
 - Riemannian optimization
 - Graph manifold learning
 - ✓ Sudah banyak paper.
-

Operator-Based Attention

- Fourier Neural Operator
 - Kernelized attention
 - Nyström attention
 - ✓ Sudah ada paper.
-

Mixture of Experts

- Sparse MoE
 - Load balancing
 - Conditional computation
 - ✓ Sangat banyak paper.
-

Optimal Transport (OT)

- OT untuk deep learning
 - Sinkhorn routing
 - Entropic regularization
 - ✓ Ada, tapi jarang digabung dengan MoE secara fundamental.
-

Stochastic Depth / Neural SDE

- Neural ODE
- Neural SDE

- Continuous-depth models
- ✓ Ada.
-

Neuro-Symbolic Integration

- Differentiable logic
 - Program synthesis + Transformer
- ✓ Ada, tapi belum mainstream.
-

Hardware-Aware Transformers

- FlashAttention
 - Tensor-core aware kernels
 - KV compression
- ✓ Ada.
-

2 Jadi Apa yang Belum Ada?

Yang belum ada adalah:

Integrasi sistematis dan koheren dari semua itu menjadi satu paradigma arsitektur tunggal dengan framing matematis baru.

Paper yang ada biasanya:

- Fokus ke satu inovasi
- Atau kombinasi 2 komponen
- Jarang >3 komponen deeply integrated

Belum ada model yang:

- Geometric manifold memory
- Integral operator attention
- OT-based expert routing
- Stochastic continuous reasoning
- Hardware-co-designed execution
- Dalam satu framework matematis unified

Itu yang membuat Sentium potensial novel.

3 Realitas Akademik yang Penting

Inovasi di AI jarang berasal dari:

“Menciptakan komponen baru dari nol”

Lebih sering berasal dari:

Menggabungkan teori berbeda secara elegan dan membuktikan bahwa integrasi tersebut menghasilkan properti baru.

Contoh sejarah:

- Transformer sendiri = kombinasi attention + residual + layer norm
 - Diffusion models = kombinasi SDE + score matching
-

4 Tapi Perlu Jujur Juga

Menggabungkan semua tanpa restraint hampir pasti:

- Tidak stabil
- Tidak trainable
- Tidak publishable

Karena novelty ≠ complexity.

Paper kuat biasanya:

- 1 core idea
 - 1 strong theoretical argument
 - 1 strong empirical result
-

5 Jadi Jawaban Final

Apakah semua yang kita bahas ada papernya?

✓ Ya, masing-masing komponen ada literturnya.

✗ Tidak, blueprint Sentium sebagai unified paradigm belum ada dalam bentuk itu.

Itulah ruang novelty Anda.

Kalau Anda mau langkah berikutnya yang lebih serius:

Saya bisa bantu:

- Mapping novelty gap secara formal
- Identifikasi kemungkinan reviewer attack
- Menentukan mana bagian yang benar-benar publishable
- Atau menyederhanakan Sentium agar realistik dan high-impact

Kita masuk ke strategi akademik sekarang?

Cari specific papers nya.

Berikut **contoh paper dan sumber akademik spesifik** untuk berbagai komponen yang relevan dengan arsitektur Sentium, termasuk hyperbolic geometry, operator learning, optimal transport, dan variasi Transformer. Ini bisa Anda jadikan referensi awal untuk literature review, sitasi, atau landasan teoretis.

Core Transformer Foundation

1. **Attention Is All You Need** – Vaswani et al., 2017 (NeurIPS)
Paper klasik yang memperkenalkan Transformer. [\[Wikipedia\]](#)
-

Operator / Neural Operator Models

2. **Transformers as neural operators for solutions of differential equations with finite regularity** – Shih et al., *Computer Methods in Applied Mechanics and Engineering* (2025)
Bukti transformator sebagai operator universal dalam operator learning. [\[ScienceDirect\]](#)
 3. **DeepONet: Learning nonlinear operators for identifying differential equations** – Lu Lu, Jin, Karniadakis, 2019
Neural operator klasik yang mendasari operator learning. [\[arXiv\]](#)
 4. **The Fourier Spectral Transformer for efficient and generalizable nonlinear PDEs** – Li et al., 2025
Transformer dengan pendekatan spectral untuk PDE. [\[ScienceDirect\]](#)
-

Hyperbolic & Geometric Transformers

5. **Exploring Efficient Transformer Fully in Hyperbolic Space** – Yang (Hypformer), 2024
Transformer yang sepenuhnya di ruang hiperbolik, termasuk linear self-attention skala besar. [\[arXiv\]](#)
6. **THG: Transformer with Hyperbolic Geometry** – Liu & Xu, 2021
Penggabungan ruang Euclidean + Hyperbolic dalam Transformer. [\[arXiv\]](#)

-
- 7. **HyLiFormer: Hyperbolic Linear Attention for Skeleton-based Action** – Li et al., 2025
Transformer dengan hyperbolic linear attention untuk efisiensi komputasi. [\[arXiv\]](#)
 - 8. **HVT: A Comprehensive Vision Framework for Learning in Non-Euclidean Space** – Fein-Ashley et al., 2024
Transformers yang memperluas representasi non-Euclidean. [\[arXiv\]](#)
 - 9. **Hyperbolic Deep Learning for Foundation Models: A Survey**, 2025
Survey komprehensif hyperbolic representation learning dalam foundation models. [\[arXiv\]](#)
-

Optimal Transport & Attention

- 10. **ESPFormer: Doubly-Stochastic Attention with Expected Sliced Transport Plans** – Shahbazi et al., 2025
Integrasi optimal transport untuk attention balance. [\[arXiv\]](#)
 - 11. **Sinkformers: Transformers with Doubly Stochastic Attention** – Sander et al., 2021
Attention layer dengan normalisasi Sinkhorn (doubly stochastic). [\[arXiv\]](#)
 - 12. **Transformers for Tabular Data: A Training Perspective of Self-Attention via Optimal Transport** – Candelieri & Quadrio, 2025
Analisis self-attention via optimal transport. [\[arXiv\]](#)
 - 13. **Provable optimal transport with transformers** – Daneshmand et al., 2024
Teori yang menjelaskan self-attention sebagai bentuk regularized OT. [\[arXiv\]](#)
 - 14. **Transformer Fusion with Optimal Transport** – Imlfeld et al., ICLR 2024
Fusi model Transformer via Optimal Transport alignment. [\[ICLR Proceedings\]](#)
-

Continuous-time / Stochastic Modeling

- 15. **The Shaped Transformer: Attention Models in the Infinite Depth-and-Width Limit** – Noci et al., 2023
Analisis Transformer melalui SDE dalam limit tak terhingga dan stabilitas. [\[arXiv\]](#)
 - 16. **OT-Transformer: A Continuous-time Transformer with Optimal Transport Regularization** – Kan, Li & Osher, 2025
Formulasi continuous-time Transformer terregularisasi OT. [\[ResearchGate\]](#)
-

Multimodal & Structure-aware Attention

- 17. **Multimodal Optimal Transport-based Co-Attention Transformer** – Xu et al., ICCV 2023
Co-attention multimodal dengan optimal transport. [\[CVF Open Access\]](#)
-

Reference Lists & Notes

📌 Banyak repository *awesome* yang mengumpulkan paper hyperbolic representation learning, berguna untuk riset lanjutan:

- **Awesome Hyperbolic Representation and Deep Learning** (GitHub) — kumpulan paper terkait ruang hiperbolik dan deep learning. [\[GitHub\]](#)
-

Tip Riset Lanjutan

Untuk membangun literatur review yang kuat, Anda perlu mencakup:

- **Transformer & attention dasar:** *Attention Is All You Need* (2017). [\[Wikipedia\]](#)
 - **Operator learning & neural operators:** *DeepONet & Fourier Neural Operator*. [\[arXiv+1\]](#)
 - **Ruang non-Euclidean & manifold learning:** Hyperbolic Transformer varian. [\[arXiv\]](#)
 - **Transport / optimization view of attention:** OT-integrated Transformer. [\[arXiv+1\]](#)
 - **Continuous-time / SDE-based frameworks:** Shaped Transformer & OT-Transformer. [\[arXiv+1\]](#)
 - **Multimodal & structured alignment:** OT-based multimodal attention. [\[CVF Open Access\]](#)
-

Jika Anda mau, saya bisa bantu:

- **susun sitasi lengkap (BibTeX)** untuk semua paper di atas
- **template survey paper** berdasarkan komponen ini
- atau **list prioritas** yang paling penting untuk research Sentium Anda

Mau lanjut ke salah satu dari itu?

Situs lengkap.

Berikut adalah **situs lengkap dalam format BibTeX** untuk sejumlah paper spesifik yang relevan dengan komponen-komponen yang dibahas sebelumnya: optimal transport dalam Transformer, hyperbolic Transformer, dan neural operator related Transformer.

BibTeX: Foundational / Transformer & Transport

```
bibtex@inproceedings{sander2022sinkformers,  
    title = {Sinkformers: Transformers with Doubly Stochastic Attention},  
    author = {Sander, Michael E. and Ablin, Pierre and Blondel, Mathieu and Peyré{'e}, Gabriel},  
    booktitle = {Proceedings of The 25th International Conference on Artificial Intelligence and Statistics},  
    series = {Proceedings of Machine Learning Research},  
    volume = {151},  
    pages = {3515--3530},  
    year = {2022},  
    publisher = {PMLR},  
    url = {https://proceedings.mlr.press/v151/sander22a.html}  
}
```

Attention dengan normalisasi Sinkhorn untuk membuat attention matrix doubly stochastic. [Proceedings of Machine Learning Research]

```
bibtex@article{daneshmand2024provable,
  title = {Provable Optimal Transport with Transformers: The Essence of Depth and Prompt Engineering},
  author = {Daneshmand, Hadi},
  journal = {arXiv},
  year = {2024},
  url = {https://arxiv.org/abs/2410.19931}
}
```

Analisis kemampuan Transformer untuk optimal transport dengan batas aproksimasi eksplisit. [arXiv]

Code

```
@inproceedings{vaswani2017attention,
  title={Attention Is All You Need},
  author={Vaswani, Ashish and Shazeer, Noam and Parmar, Niki and Uszkoreit, Jakob and Jones, Llion and Gomez, Aidan
N and Kaiser, {\L}ukasz and Polosukhin, Illia},
  booktitle={Advances in Neural Information Processing Systems},
  volume={30},
  year={2017},
  url={https://papers.nips.cc/paper/2017/hash/3f5ee243547dee91fdb053c1c4a845aa-Abstract.html}
}
```

Makalah asli Transformer. [Wikipedia]

```
Code@article{chen2024pit,
  title={Positional Knowledge is All You Need: Position-induced Transformer (PiT) for Operator Learning},
  author={Chen, Junfeng and Wu, Kailiang},
  journal={arXiv},
  year={2024},
  url={https://arxiv.org/abs/2405.09285}
}
```

Transformer dengan posisi-attention untuk operator learning. [arXiv]

Code

```
## 🚧 BibTeX: Hyperbolic & Geometric Transformers
```

```
```bibtex
@inproceedings{yang2024hypformer,
 title={Hypformer: Exploring Efficient Hyperbolic Transformer Fully in Hyperbolic Space},
 author={Yang, Menglin and Verma, Harshit and Zhang, Delvin Ce and Liu, Jiahong and King, Irwin and Ying, Rex},
 booktitle={KDD 2024},
 year={2024},
 url={https://paperswithcode.com/paper/hypformer-exploring-efficient-hyperbolic}
}
```

*Hyperbolic Transformer dengan mekanisme hyperbolic self-attention linier.* [Papers with Code]

```
bibtex@inproceedings{yang2025hgformer,
 title={Hgformer: Hyperbolic Graph Transformer for Collaborative Filtering},
 author={Yang, Xin and Li, Xingrun and Chang, Heng and Jinze, Yang and Yang, Xihong and Tao, Shengyu and Shigeno,
Maiko and Chang, Ningkang and Wang, Junfeng and Yin, Dawei and Min, Erxue},
 booktitle={Proceedings of the 42nd International Conference on Machine Learning},
 series={Proceedings of Machine Learning Research},
 volume={267},
 pages={70813--70832},
 year={2025},
 publisher={PMLR},
 url={https://proceedings.mlr.press/v267/yang250.html}
}
```

*Hyperbolic Graph Transformer yang memanfaatkan manifold global dengan cross-attention hyperbolic.* [Proceedings of Machine Learning Research]

---

## Misc (Relevant Emerging Operator / Linear Attention)

```
bibtex@article{hu2025transolver,
 title={Transolver is a Linear Transformer: Revisiting Physics-Attention through the Lens of Linear Attention},
 author={Hu, Wenjie and Liu, Sidun and Qiao, Peng and Sun, Zhenglun and Dou, Yong},
 journal={arXiv},
 year={2025},
 url={https://arxiv.org/abs/2511.06294}
}
```

*Linear Transformer untuk Neural Operator & PDE dengan pendekatan “physics attention”.* [\[arXiv\]](#)

```
bibtex@article{zhong2025lano,
 title={Efficient High-Accuracy PDEs Solver with the Linear Attention Neural Operator},
 author={Zhong, Ming and Yan, Zhenya},
 journal={arXiv},
 year={2025},
 url={https://arxiv.org/abs/2510.16816}
}
```

*Neural operator dengan linear attention scalable dan akurat.* [\[arXiv\]](#)

Code

---

## 🔍 Struktur Lainnya Teoretis / Survey

> (Anda bisa tambahkan paper lain seperti \*Fourier Neural Operator\*, \*Linformer\*, \*Reformer\*, \*Sparse transformers\*, \*Hyperbolic representation surveys\*, atau \*operator learning surveys\* jika perlu sitasi lebih luas.)

---

 Jika Anda ingin versi \*\*APA\*\* atau \*\*MLA\*\* untuk setiap paper di atas, saya bisa konversinya.

 Jika Anda ingin daftar referensi ini dibagi menurut \*sub-topik\* (mis. Optimal Transport, Hyperbolic Geometry, Neural Operators), saya juga bisa buatkan daftar terstruktur lengkap.