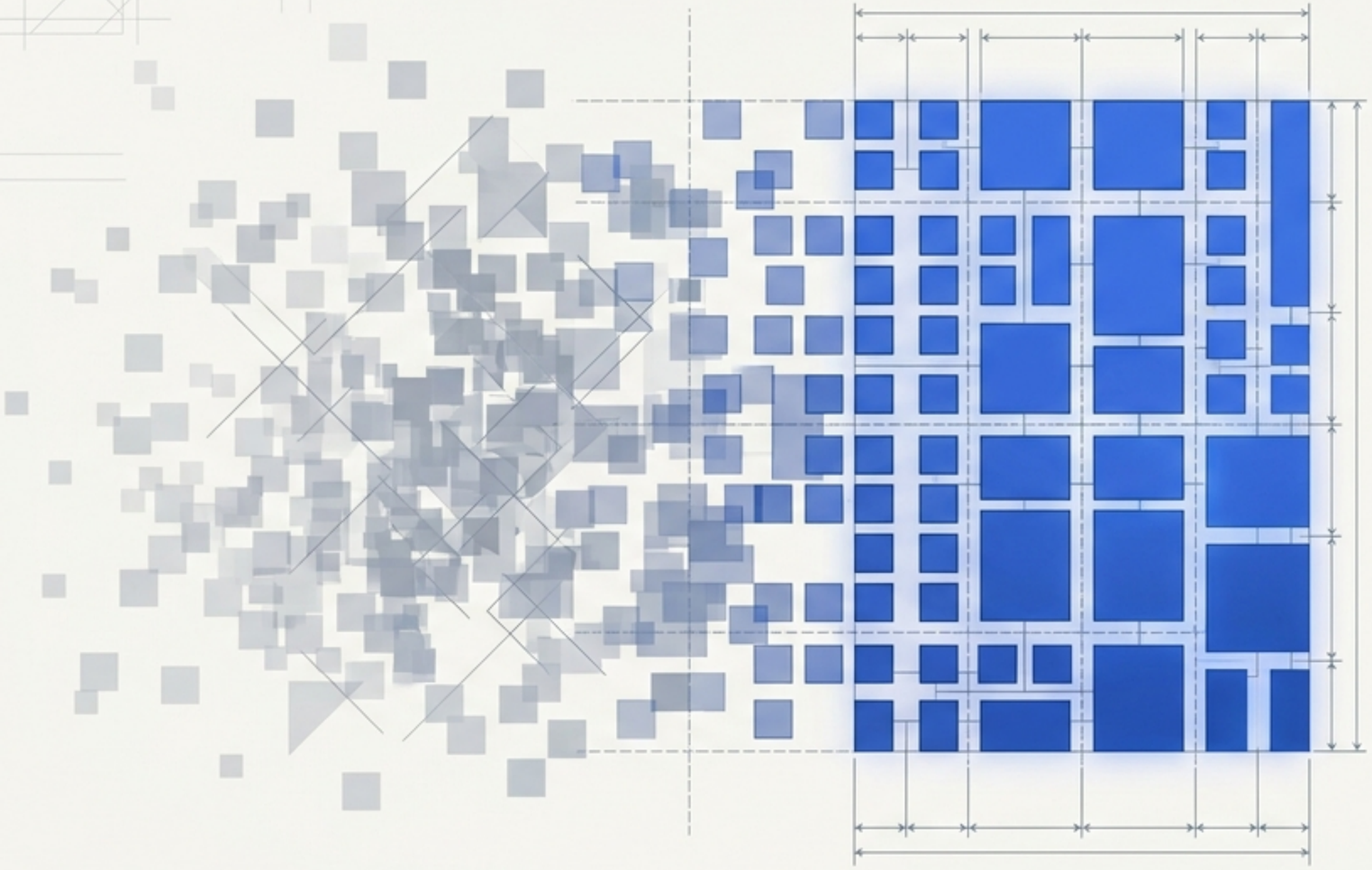


The Resolving Canvas

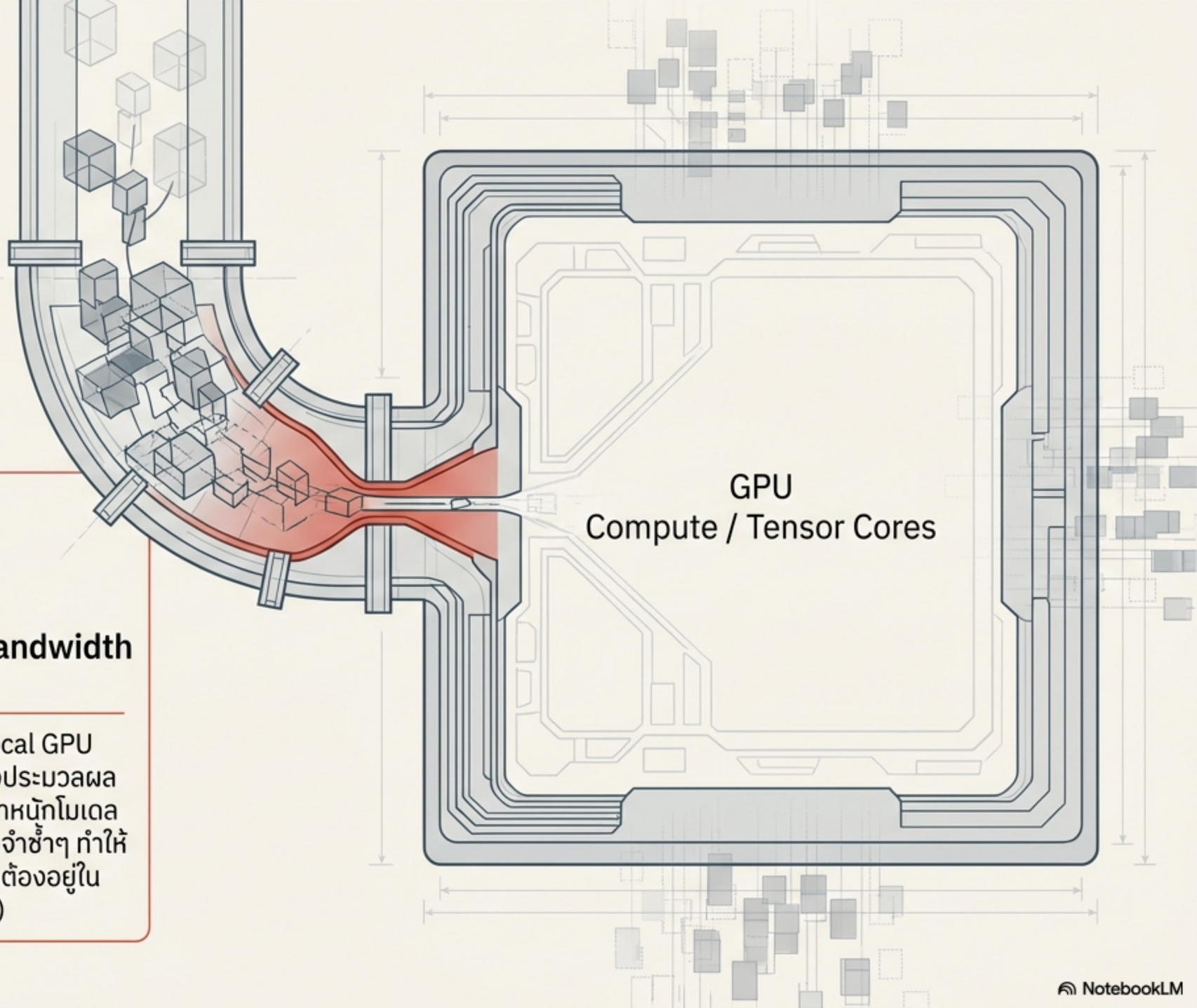
Technical Explainer & Architecture Teardown

DiffusionGemma: การก้าวกระโดดสู่ยุค Text Diffusion

โมเดล 26B MoE จาก Google ที่เปลี่ยนกระบวนทัศน์จากการสร้างข้อความทีละโทเค็น สู่การประมวลผลแบบขนาน (Parallel Generation)

NotebookLM

ปัญหาคอขวด ของการประมวลเดิม



โครงสร้างแบบ Sequential

โมเดลภาษาแบบ Autoregressive สร้างทีละโทเค็นจากซ้ายไปขวา ต้องพึ่งพาสิ่งที่สร้างมาก่อนหน้าเสมอ

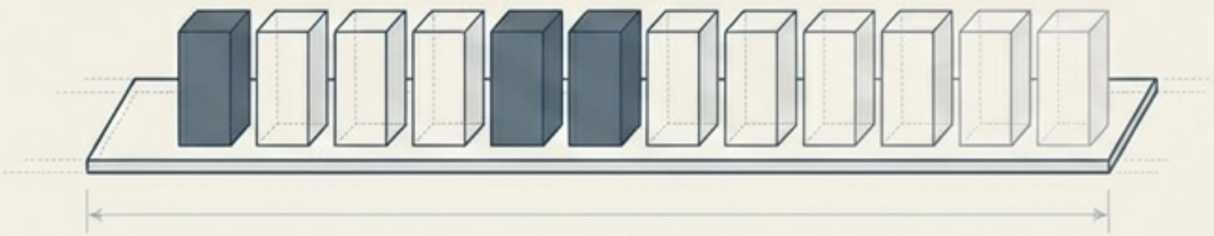
Memory Bandwidth คือตัวถ่วง

ในการรันบน Local GPU ปัญหาไม่ใช่พลังประมวลผล แต่เป็นการดึงน้ำหนักโมเดลจากหน่วยความจำช้าๆ ทำให้ Tensor Cores ต้องอยู่ในสถานะรอ (Idle)

NotebookLM

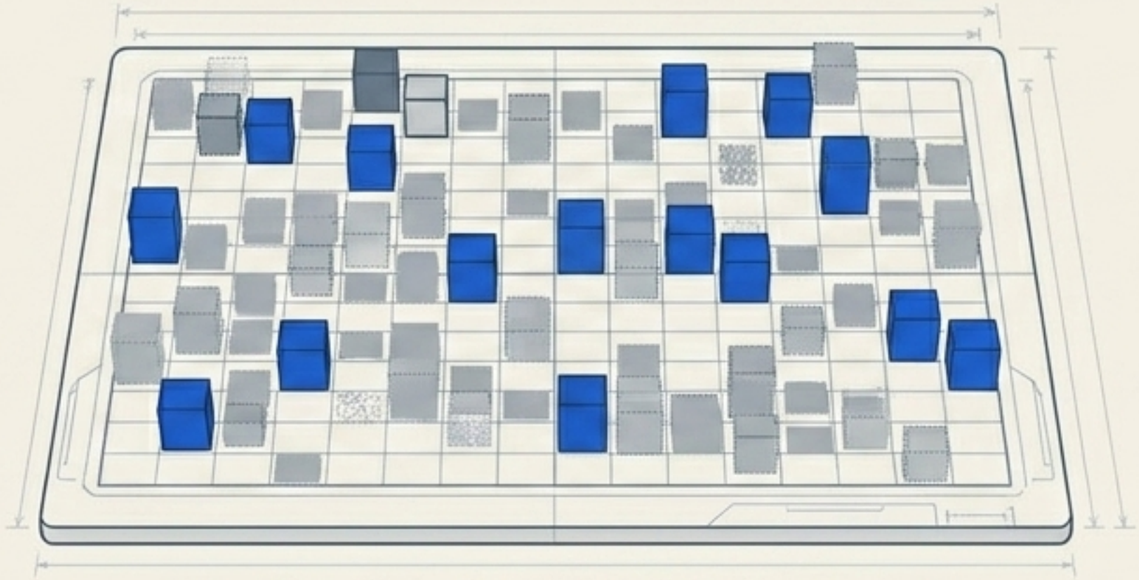
จาก Sequential สู่ Parallel: ความต่างของกระบวนทัศน์

Autoregressive



วางทีละบล็อก (Token-by-Token)
ประมวลผลเป็นเส้นตรงแบบทางเดียว

Diffusion



สร้างทั้งบล็อกพร้อมกัน (Parallel)
ปรับแก้โทเค็นทั้งหมดไปพร้อมๆ กันบนผ้าใบ

ประสิทธิภาพ
สร้างข้อความได้เร็วกว่าสูงสุด 4 เท่า
(Up to 4x Faster) บน Dedicated GPU

NotebookLM

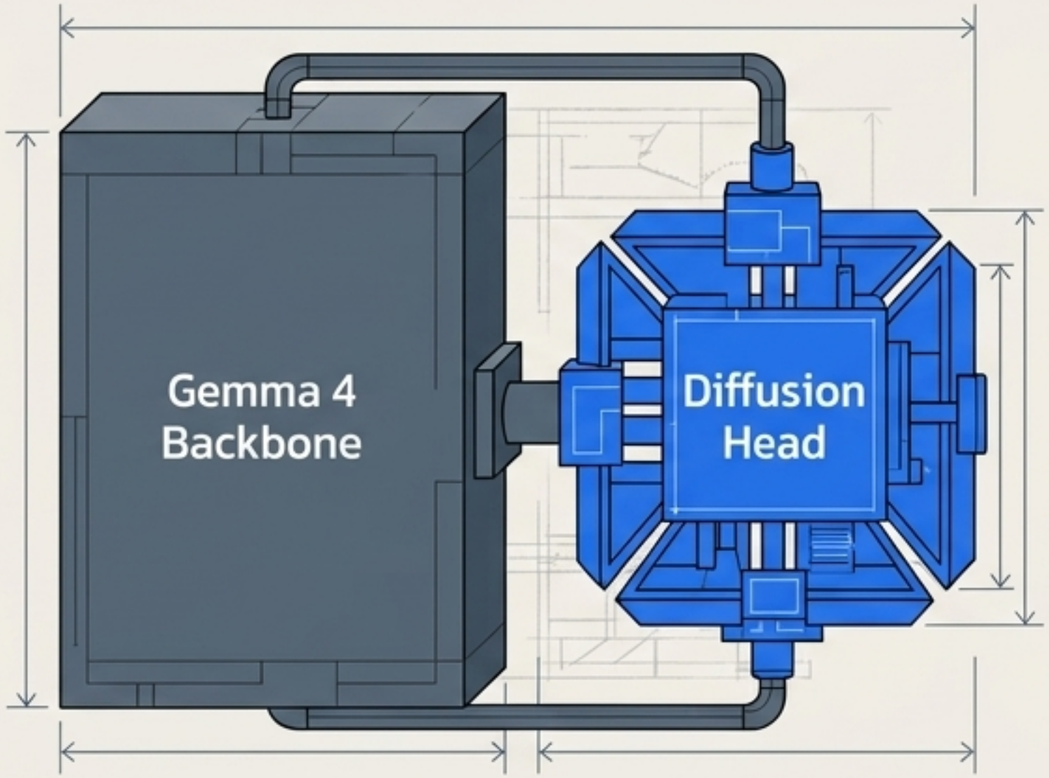
สถาปัตยกรรมระดับ 26B บนพื้นฐาน Gemma 4

Parameters

26B MoE (Mixture of Experts) / ใช้งานจริงเพียง 3.8B Active Parameters ต่อรอบ

Context Window

รองรับบริบทความยาว 256K Tokens และสนับสนุนกว่า 140+ ภาษา



Inputs & Outputs

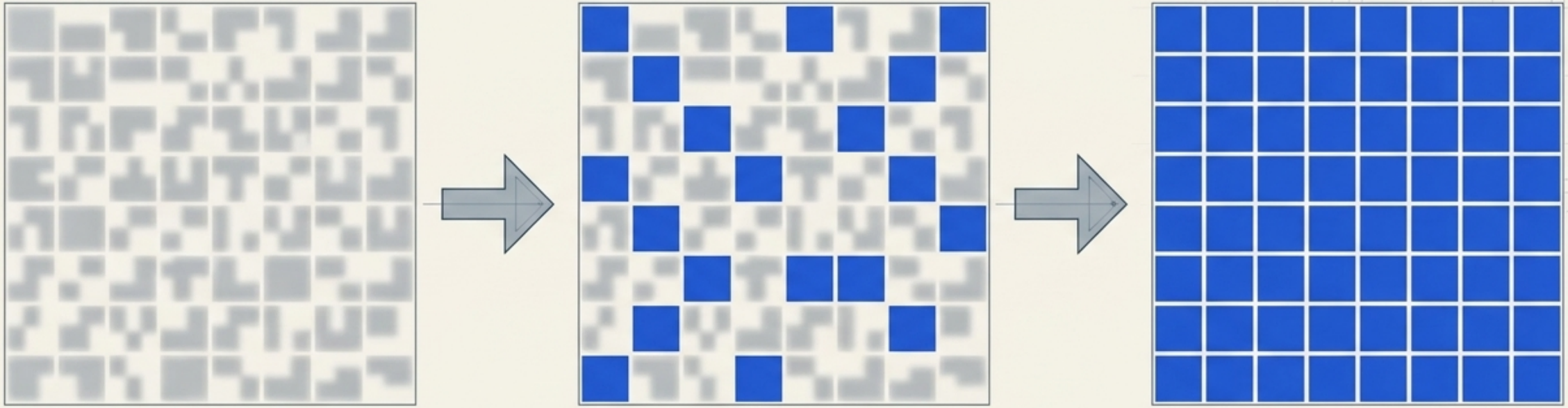
Multimodal Input (ข้อความ, รูปภาพ, วิดีโอ) แปลงสู่ผลลัพธ์เป็นข้อความ

Open Weights

Apache 2.0 (พร้อมสำหรับการนำไปพัฒนาต่ออย่างอิสระ)

NotebookLM

Uniform State Diffusion ทำงานอย่างไร?



1. The Noise Canvas

โมเดลเริ่มต้นด้วยผ้าใบที่เต็มไปด้วยโทเค็นแบบสุ่มความยาว 256 โทเค็น

2. Iterative Refinement

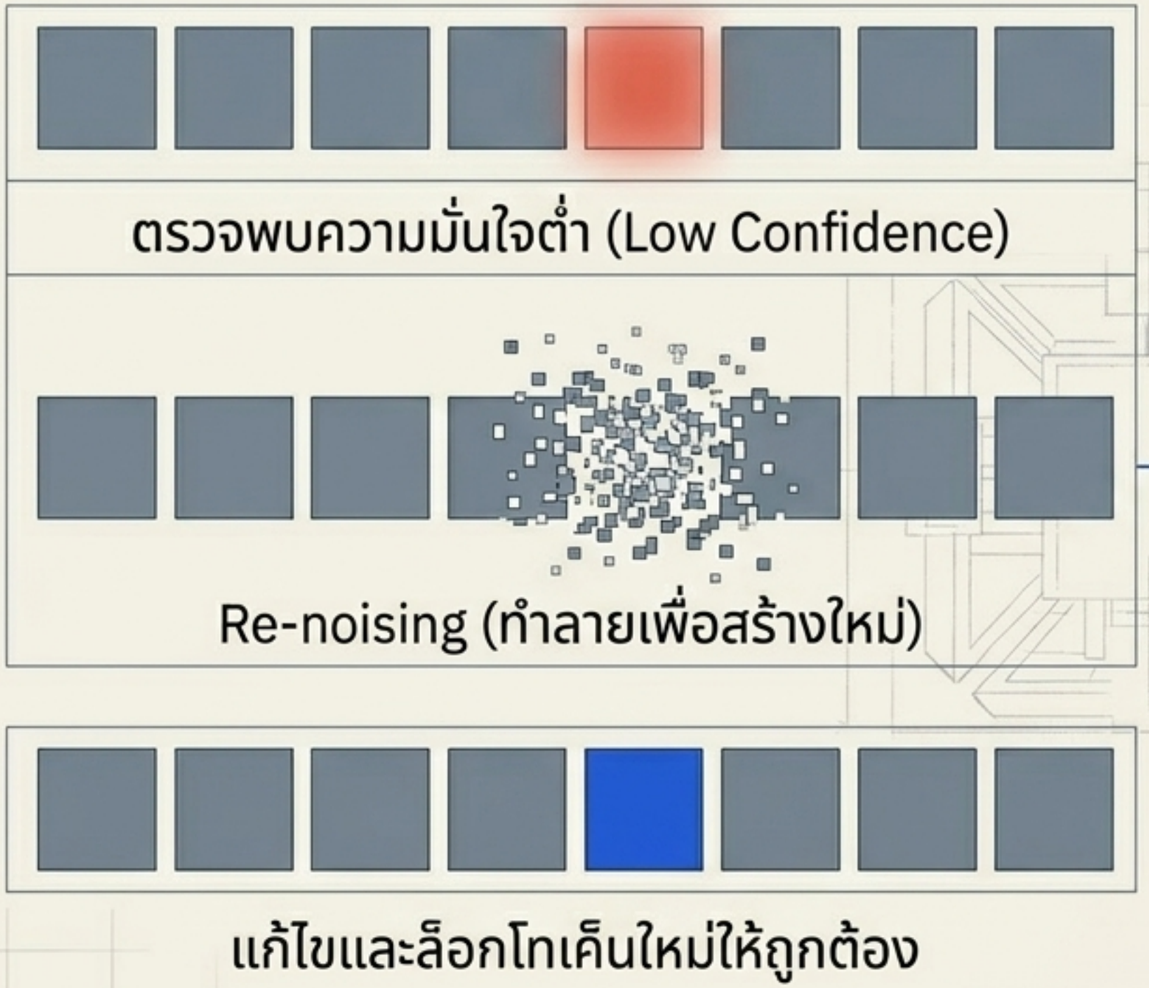
กวาดสายตาบนผ้าใบ ล็อกโทเค็นที่มีความมั่นใจสูง ยืนยันเฉลี่ย 15-20 โทเค็น ต่อหนึ่ง Forward Pass

3. Convergence

ข้อความทั้งหมดจะถูกปรับแต่งจนสมบูรณ์ชัดเจนในรูปแบบ Parallel

NotebookLM

พลังพิเศษ: การแก้ไขตัวเองแบบ Real-time (Self-Correction)



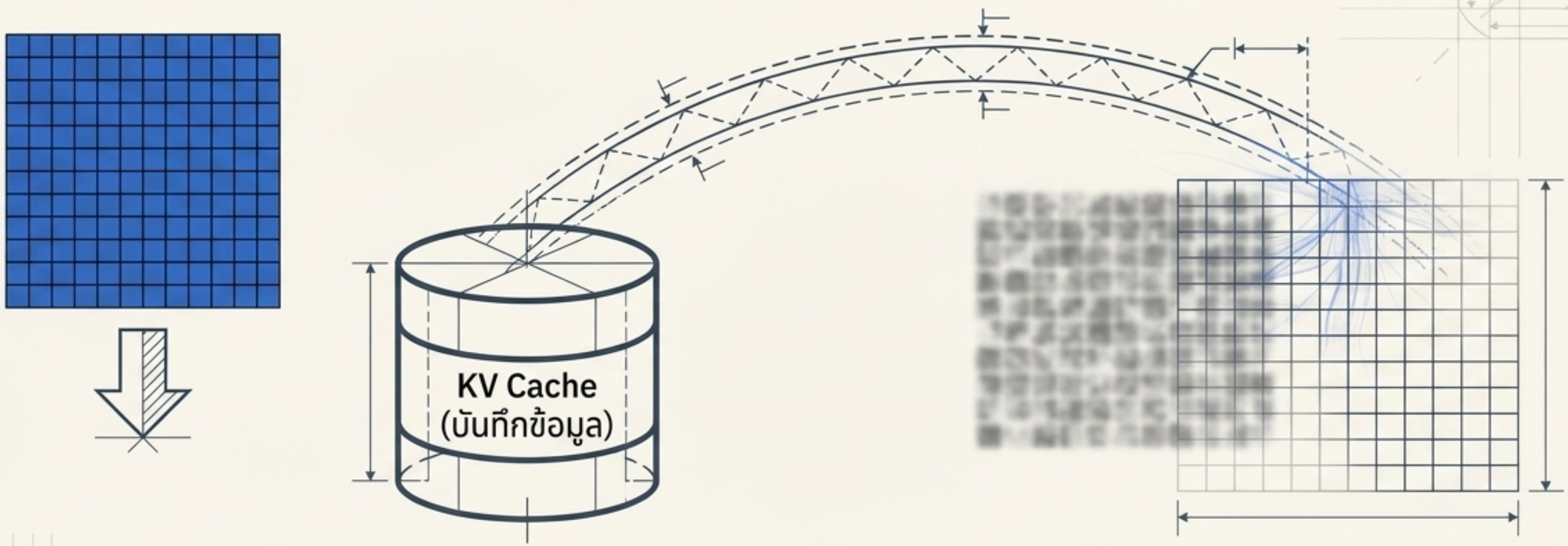
Bidirectional Attention

ทุกโทเค็นสามารถสังเกตโทเค็นอื่นๆ ได้ทุกทิศทาง ระบบจึงสามารถย้อนกลับไปปแก้ไขสิ่งที่ผิดพลาดได้ ซึ่งโมเดลแบบเก่า (Autoregressive) ทำไม่ได้ พราะโทเค็นถูกสร้างแล้วจะตายตัวทันที

NotebookLM

Block Autoregressive Diffusion

การจัดการข้อความยาวที่เกินขนาดผ้าใบ (Long Context)



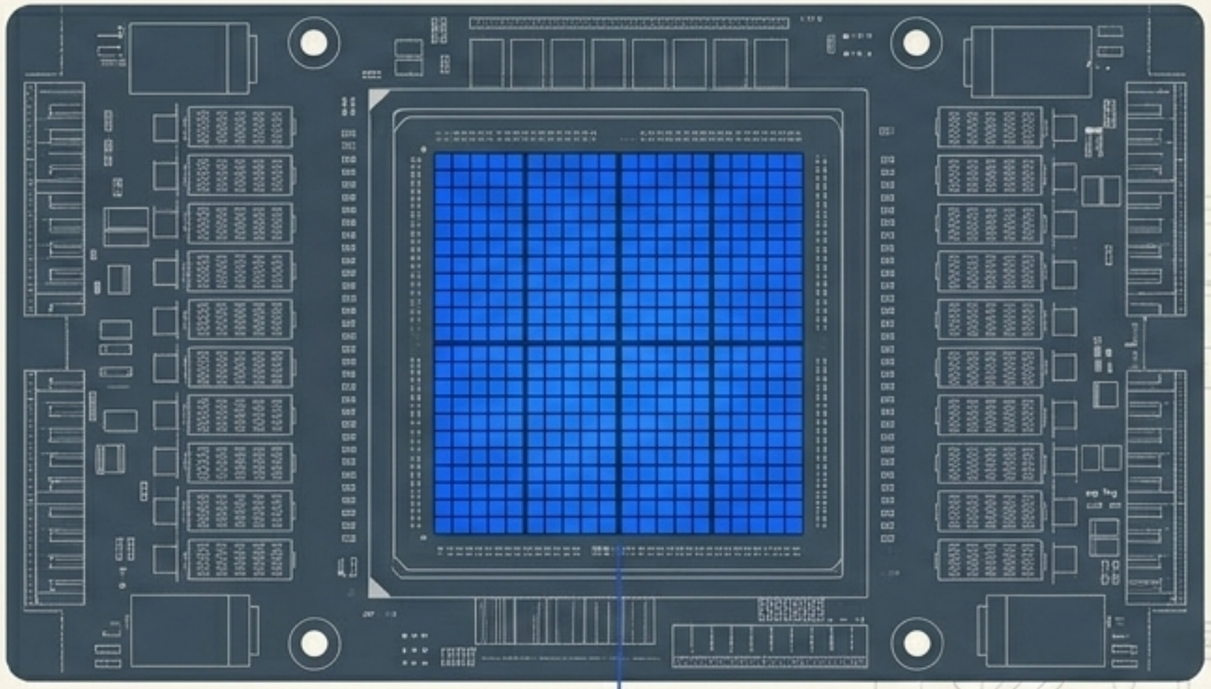
Step 1:
เมื่อผ้าใบขนาด 256 โทเค็นแรกถูกสร้างจนสมบูรณ์

Step 2:
ข้อมูลบริบททั้งหมดจะถูกจัดเก็บลงใน KV Cache

Step 3:
สร้างแผ่นใหม่โดยอ้างอิงประวัติจากบล็อกก่อนหน้า
ผสานความเร็วแบบ Parallel และความเสถียรแบบลำดับขั้น

NotebookLM

ย้ายคอขวดจาก Memory สู่ Compute



Full Tensor Core Utilization

Hardware Insight: การประมวลผลผ้าใบ 256 โทเค็นพร้อมกัน เป็นการป้อนปริมาณงานให้ Tensor Cores ทำงานเต็มประสิทธิภาพ 100%

Accessibility: สามารถรันได้ภายในหน่วยความจำ 18GB VRAM (ผ่านการ Quantize แบบ NVFP4)

NotebookLM

ตัดสินใจเลือก: DiffusionGemma vs. Standard Gemma 4

คุณสมบัติ	DiffusionGemma (26B)	Standard Gemma 4
วิธีสร้าง (Generation)	**Parallel Diffusion**	Autoregressive
ข้อจำกัดฮาร์ดแวร์	**Compute-bound**	Memory-bandwidth
ประเภท Attention	**Bidirectional (สองทิศทาง)**	Causal (ย้อนหลังเท่านั้น)
การแก้ไขตัวเอง	**มี** (ผ่าน Re-noising)	ไม่มี (Commit ตายตัว)
ความเร็วบน GPU	**เร็วกว่าสูงสุด 4x**	Baseline มาตรฐาน
รูปแบบงานที่เหมาะสม	**Local, ตอบสนองไวแบบโต้ตอบ**	High-QPS Cloud Serving

NotebookLM

ข้อได้เปรียบในงานที่มีข้อจำกัดซับซ้อน (Constrained Generation)

Autoregressive
(ความแม่นยำ 0%)
- ตัดสินใจพลาดแล้วแก้ไม่ได้

	2	1	1	9	3	4	7	
7	3	6	6	4	9	2		5
4	7	5	2	1	6		9	1
5	1	2	3	6		7	8	6
7	8	1	4		8	4	3	7
3	6	7		2	9	5	6	4
8	7		7	4	6	2	1	5
		3	5	2	7	1	4	6
	6	1	9	3	4	8	3	7

Fine-tuned Diffusion
(ความแม่นยำ 80%) -
ประเมินเงื่อนไขภาพรวมได้

Insight: การมองเห็นบริบทแบบสองทิศทาง (Bidirectional) ช่วยให้โมเดลสามารถหยุดพักและประเมินเงื่อนไขทั้งหมดได้ก่อนจะสิ้นสุดการทำงาน เหมาะกับโจทย์คณิตศาสตร์ โค้ดดิ้ง และตรรกะ

NotebookLM

กรณีการใช้งานที่เหมาะสมที่สุด (Optimal Use Cases)

In-line Editing & Code Infilling

โครงสร้างข้อความแบบ Non-linear ได้รับประโยชน์สูงสุดจากการอ่านบริบทแบบสองทิศทาง

Local Rapid Iteration

ความหน่วงต่ำ (Low Latency) เหมาะสำหรับงานพัฒนาและทดสอบโค้ดบนเครื่องนักพัฒนาคนเดียว

OCR & Document Parsing

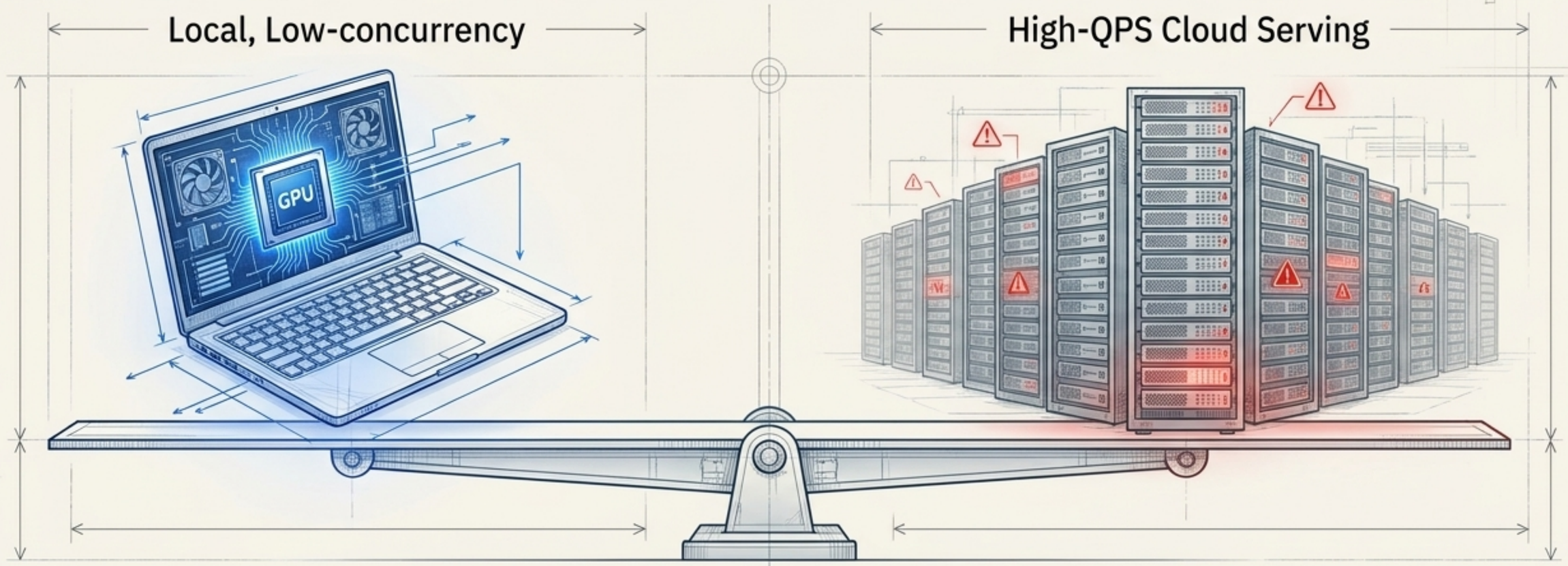
รองรับการป้อนข้อมูลภาพ (Multimodal) และมี Context Window ขนาดใหญ่ถึง 256K

Agentic Workflows

รองรับการเรียกใช้เครื่องมือ (Tool Calling) ภายในสถาปัตยกรรม Local Agents

NotebookLM

ข้อควรระวัง: สภาพแวดล้อมคือตัวแปรสำคัญ



สภาพแวดล้อมในอุดมคติ โมเดลสามารถใช้
Compute ของ GPU เดี่ยวได้อย่างเต็มประสิทธิภาพ
สร้างประสบการณ์แบบโต้ตอบ (Interactive) ได้ดีเยี่ยม

ในระบบที่มีผู้ใช้งานพร้อมกันจำนวนมาก
โมเดลแบบเดิม (Autoregressive) จัดการทรัพยากรคุ้มค่ากว่า
การใช้ Diffusion ในจุดนี้อาจเพิ่มต้นทุนโดยไม่จำเป็น

NotebookLM

การสนับสนุนจาก Ecosystem ตั้งแต่วันแรก

DiffusionGemma 26B

Open Source Weights

ดาวน์โหลดได้ทันทีบน Hugging Face (Apache 2.0)

Deployment

พร้อมใช้งานทันทีบน Google Cloud Model Garden และ NVIDIA NIM

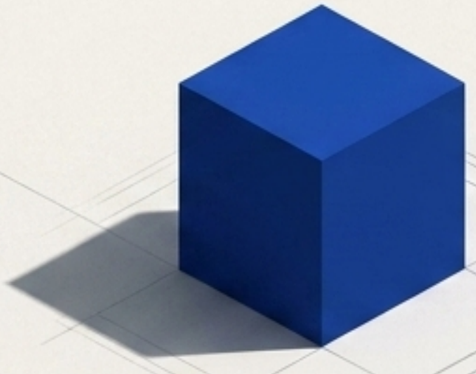
Frameworks

โมเดล Diffusion LLM ตัวแรกที่รองรับแบบ Native บน vLLM

Integrations

ใช้งานได้ผ่าน Transformers, MLX, Unsloth และ Fine-tune ผ่าน NeMo

NotebookLM



ทิศทางใหม่ของการประมวลผล AI ระดับ Local

DiffusionGemma พิสูจน์ให้เห็นว่า อนาคตของ AI ไม่ได้ขึ้นอยู่กับการลดขนาดโมเดลเพียงอย่างเดียว แต่คือการเปลี่ยนวิธีถอดรหัสของ GPU โดยสิ้นเชิง

ด้วยการปลดล็อกข้อจำกัดทาง Memory สู่การใช้ Compute เต็มศักยภาพ Text Diffusion เปิดประตูสู่ยุคใหม่ของแอปพลิเคชัน Real-time บน Consumer Hardware อย่างแท้จริง

NotebookLM