In-Memory Computing (Automotive)

Disruptive Applications in Autonomous Driving

null V1.0

Preface: redefining computing power

Traditional TOPS: meaning and calculation

In the modern AI chip performance evaluation system, TOPS (Tera Operations Per Second) has become the most mainstream computing power indicator. Whether it is NVIDIA's GPU, Qualcomm's NPU, or Horizon and Black Sesame's autonomous driving chips, TOPS is used as the performance benchmark.

Mathematical definition of TOPS

The calculation of TOPS is based on the MAC (Multiply-Accumulate) unit inside the chip, which is the core component of digital computing:

Calculation formula:

TOPS = (2 × number of MAC units × operating frequency) / 10^{12}

Formula explanation:

• Number of MAC units: The number of multiply-accumulate operation units working in parallel within the chip (for example, NVIDIA Orin-X has 2048 INT8 MAC units).

• Operating frequency: The operating frequency of the MAC unit (eg 1.6 GHz).

• Coefficient "2": A MAC unit completes one multiplication + one addition in one clock cycle, which is counted as 2 operations.

• Divide by 10^{12}: Convert operations per second (OPS) to TOPS units.

Example calculation (NVIDIA Orin-X):

• Number of MAC units: 2048 (INT8 precision)

• Operating frequency: 1.6 GHz

• TOPS = (2 × 2048 × 1.6 × 10^{9}) / 10^{12} = 6.55 TOPS (single core)

• Orin-X has multiple computing cores with a total computing power of 254 TOPS (INT8)

Physical implementation of MAC units

The traditional digital MAC unit is implemented based on binary logic gate circuits:

Input

• A (weight): 8-bit integer, such as 01101100

• B (activation value): 8-bit integer, such as 10110011

Processing Process

Step 1: Multiplier (Multiplier)

• Expand to 8×8=64 AND gates (AND Gate)

• Compress partial products via Wallace Tree

• Output 16-bit product result

Step 2: Adder (Adder)

• Accumulate the product result into the accumulator (Accumulator)

• Use Carry-Lookahead Adder

• Output the final accumulated value

Power consumption characteristics:

• Logic gate flip power consumption: ~10 pJ* per MAC operation (INT8)

• Accumulator memory access power consumption: about 20 pJ (read-modify-write cycle)

• Total power consumption: ~30 pJ/MAC

* Note: pJ (picojoule, picojoule) is an energy unit, 1 pJ = 10^{-12}Joule, used to measure tiny energy consumption at the chip level

Key features:

• Numerical precision calculations: the results are completely certain and there is no accumulation of errors.

• High energy consumption: Each operation requires millions of transistors to flip states.

• Limited by clock frequency: Improving TOPS requires higher frequency or more MAC units, resulting in a linear increase in chip area and power consumption.

The data-movement tax of von Neumann architectures

Even with powerful MAC cell arrays, traditional architectures still face fundamental bottlenecks:

Data flow path (taking autonomous driving inference as an example):

1. Model weights are stored in UFS/SSD (bandwidth ~3.6 GB/s)

2. Load the weights into DRAM (LPDDR5, bandwidth 204 GB/s)

3. DRAM → NPU Cache (bandwidth ~1 TB/s)

4. Cache → MAC array (perform calculations)

5. Write intermediate results back to DRAM

6. The final result is written back to storage

[Energy consumption in each link]

• UFS → DRAM: ~35 pJ/bit

• DRAM → Cache: about 5 pJ/bit

• Cache → MAC: about 1 pJ/bit

• MAC calculation: ~30 pJ/MAC

• Total energy consumption: data handling accounts for 60–90%

The vicious circle of "moving tax":

• The amount of AI model parameters exploded: from 10B (GPT-2) → 175B (GPT-3) → 1750B (GPT-4), and the amount of weight data increased 175 times.

• Memory bandwidth wall: DRAM bandwidth increases only 2–3 times (from 68 GB/s in DDR4 → 204 GB/s in LPDDR5).

• Energy efficiency bottleneck: In the autonomous driving scenario, 63.7% of the power consumption is used for data transfer, and only 28.5% is used for actual calculation.

Conclusion: Although traditional TOPS has high computing power, it is limited by "the data cannot be moved", and the actual effective computing power utilization is only 20-40%.

iNc: a paradigm shift to analog computation

Core principle: Ohm’s law + Kirchhoff’s law

iNc technology adopts Analog In-Memory Computing, completely breaking out of the "digital MAC + data transfer" framework:

Physical implementation (based on 3D NAND Flash array):

Matrix multiplication: $Y = W \times X$

Physical Mapping

• W (weight matrix): stored in the conductance (G) of the NAND memory cell

→ Adjust the conductance by controlling the threshold voltage (Vth) of the Flash unit

• X (input vector): converted to voltage (V) applied to the word line (Word Line)

• Y (Output Vector): Current collected through Bit Line (I)

[Calculation process (completed in one step)]

Step 1: Apply input voltage V_1, V_2, ..., V_n to each word line

Step 2: According to Ohm’s law $I = G \times V$

→ Each Flash unit generates current $I_{ij} = G_{ij} \times V_j$

Step 3: According to Kirchhoff’s current law, the current on the bit line is automatically summed

→ $I_i = \Sigma(G_{ij} \times V_j) = (W \times X)_i$

Step 4: Convert the current into digital results via ADC (Analog-to-Digital Converter)

Key advantages

• Zero data transfer: weights are resident in Flash and calculations are completed within the array

• Extremely high degree of parallelism: M×N matrix multiplication is completed in O(1) time (vs. O(M×N) for numbers)

• Energy efficiency improvement: Avoid digital logic gate flipping, reducing power consumption by 10–100 times

• Delay reduction: Eliminate DRAM memory access delay (about 100ns) and reduce ADC conversion delay (about 1μs)

Calculated performance (512Gb 3D NAND chip):

• Vector similarity calculation (512 dimensions): 50 μs (vs digital solution 2.35 ms)

• Energy efficiency improvement: 43 times (vs CPU/GPU)

• Latency reduction: 47 times

Why traditional TOPS does not apply to iNc

Comparison of essential differences:

Dimension	Traditional digital computing (MAC)	iNc analog-domain computing
Compute unit	Digital logic gates (AND gates and adders)	Physical devices (Flash conductance and current summation)
Compute precision	Deterministic (e.g. INT8, 256 discrete values)	Continuous analog values (ADC-limited; e.g. 8-bit ADC, 256 levels)
Parallelism	Depends on MAC count (e.g. 2,048 MACs)	Full-array parallelism (M×N matrix in one step)
Source of energy efficiency	High-frequency clock driven (e.g. 1.6 GHz)	Driven by physical laws (currents add naturally)
TOPS calculation	2 × MAC count × frequency / 10^{12}	Not applicable (no MAC units or clock frequency)
Core advantage	General-purpose capability and controllable precision	Very high efficiency for data-intensive tasks

Why forcing TOPS onto iNc is misleading

1. The physical architecture is different: iNc does not have a digital MAC unit and no global clock frequency (analog circuits are continuous-time systems).

2. Different computing paradigms: iNc's "one matrix multiplication" corresponds to the "thousands of MAC operations" of the digital system, reducing the time complexity and dimensionality.

3. The optimization goals are different: TOPS optimizes "number of operations per second", and iNc optimizes "single task completion time and energy consumption".

Analogy:

• TOPS = A measure of "the number of bricks a worker moves per hour" (suitable for assembly line operations).

• iNc = Replace "manual brick moving" with "automatic conveyor belt handling" (jumping out of the evaluation dimension of "number of bricks moved").

Estimating equivalent iNc performance

Although iNc does not provide traditional TOPS values, it can calculate equivalent computing power through task completion delay:

Scenario: Vector similarity retrieval (VGGFace2 data set, 170,000 512-dimensional face feature vectors)

Traditional solution (CPU/GPU):

• Algorithm: Calculate cosine similarity for 170,000 vectors one by one (requiring 512 multiplications + 512 additions × 170,000)

• Total operations: 512 × 2 × 170,000 = 174 million operations

• Required computing power (completion time 20 seconds): 1.74×10^8 / 20s = 8.7 GOPS ≈ 10 TOPS (considering data transfer overhead)

• Actual power consumption: 100 J (including DRAM access + CPU calculation)

iNc solution (in-memory computing):

• Algorithm: Query vector is applied as voltage, 170,000 weight vectors are stored as Flash conductance, and similarity is calculated in parallel

• Latency: < 0.5 seconds

• Energy consumption: < 2.5 J (43x reduction)

• Equivalent computing power: If converted in terms of "time to complete the same task", it is equivalent to 10 TOPS × (20s / 0.5s) = 400 TOPS (but the physical architecture is completely different)

Conclusion:

The value of iNc does not lie in "how many TOPS you have", but in:

1. Extreme energy efficiency for specific tasks: for tasks such as vector retrieval, similarity matching, and Embedding search, the energy efficiency is increased by 43 times.

2. Reduced latency: Eliminate data transfer bottlenecks and reduce latency by 47 times.

3. Bandwidth bottleneck elimination: weights are resident in Flash and do not need to occupy DRAM bandwidth.

Applicable scenarios:

• Data-intensive tasks: large-scale vector retrieval, image/video feature matching, semantic search

• Bandwidth-sensitive tasks: multi-sensor fusion, BEV feature fusion, Transformer KV cache management

• Low power consumption scenarios: wearable devices, smart glasses, vehicle edge computing

• Not suitable for: general scientific computing, high-precision numerical simulation (limited by ADC accuracy, usually 8–12 bit)

Autonomous driving: from peak throughput to effective computing

The computing power dilemma of autonomous driving

The vicious cycle of traditional solutions:

[L4 autonomous driving requirements]

• 8 cameras (8MP @30fps) → 760 MB/s raw data

• 1 128-line lidar → 120 MB/s point cloud data

• 5 millimeter wave radars → 300 MB/s 4D imaging data

• Total: 1.18 GB/s sensor data stream

[Traditional architecture processing]

Step 1: Data from sensor → SoC DRAM (power consumption accounts for 40%)

Step 2: DRAM → NPU calculation (power consumption accounts for 30%)

Step 3: NPU calculation (power consumption accounts for 30%)

→ Actual effective computing power consumption accounts for < 30%, and the remaining 70% is used for data transfer

[OEMs are forced to stack computing power]

• Dual NVIDIA Orin-X (508 TOPS) → Power consumption 120W

• Actual effective computing power utilization: 20–40%

• Range impact: 10–15%

• BOM cost: > 5,000 yuan

The core contradiction: It is not that "TOPS is not enough", but "TOPS is not used" - the memory bandwidth bottleneck makes the computing power "starve to death".

How iNc overcomes the bottlenecks

Design philosophy: Move computing to where the data is, rather than moving data to computing units.

Architecture Innovation

Edge Sensor (iNc Sensor Hub):

• Built-in 64–128GB iNc eUFS per camera/radar module

• Complete feature extraction in place (first few layers of YOLO/Transformer)

• Data compression 10:1 (only feature vectors are transmitted, not original images)

• Power consumption: < 1W/module

Central Computing Platform (iNc Auto-L4):

• 512GB–1TB iNc UFS (storage model weights + historical frame features)

• Multi-sensor feature fusion is done within the NAND array (analog domain current summation)

• Works with lightweight SoC (50–100 TOPS) to handle control flow

• System power consumption: < 80W (vs traditional 150W)

[Performance improvement]

• Data transfer power consumption ratio: reduced from 63.7% to < 10%

• Effective computing power utilization: increased from 20–40% to 70–90%

• Multi-sensor fusion latency: reduced from 80–150ms to < 30ms

• End-side large model inference (35B VLA): reduced from 200ms to < 100ms

Economic Value

• BOM cost: reduced by 30–40% (reduces the need for high-end SoC + large-capacity DRAM)

• Improved energy efficiency: power consumption is reduced by 30–50% under the same performance, and the impact on cruising range is reduced from 15% to 8%

• R&D cycle: from "modular solution (2 years)" to "end-to-end model (< 1 year)"

Core conclusion: iNc in-memory computing provides a feasible technical path for autonomous driving to evolve from L3 to L5 by eliminating data transfer bottlenecks, improving computing power utilization, and reducing system power consumption, while significantly lowering the commercialization threshold.

1. Autonomous-driving computing challenges

1.1 Compute demand versus effective utilization

According to calculation data from relevant organizations, the computing power requirements for each level of autonomous driving are as follows:

Autonomous-driving level	Theoretical compute demand (TOPS, INT8)	Estimated effective utilization	Typical power
L2 (advanced driver assistance)	2–10 TOPS	60–80%	10–20W
L2+ (enhanced driver assistance)	10–30 TOPS	40–60%	20–40W
L3 (conditional automation)	30–60 TOPS	30–50%	40–80W
L4 (high automation)	200–500 TOPS	20–40%	80–150W
L5 (full automation)	1000–2000+ TOPS	Estimated < 30%	150–300W

Key findings

1. Computing power utilization decreases with complexity: Although the L4-level system is equipped with a peak computing power of 500+ TOPS, it is limited by factors such as memory bandwidth bottlenecks, data transfer overhead, and multi-sensor timing out-of-synchronization. The actual effective utilization rate is only 20–40%.

2. Data transfer becomes an energy efficiency black hole: According to the power consumption deconstruction analysis of NVIDIA Orin-X, 63.7% of the power consumption is used for data transfer between DRAM and computing units (DDR5 ↔ NPU), and the proportion of power consumption actually used for MAC operations is < 30%.

3. The cycle of "computing power inflation": To make up for low utilization, OEMs are forced to stack more chips (such as dual/quad Orin-X), causing cost, power consumption, and heat dissipation issues to spiral.

1.2 The memory-bandwidth wall in autonomous driving

Data flow characteristics of typical L4 systems:

Data source	Resolution / sample rate	Raw-data stream	Feature-data stream
Eight cameras	8 megapixels @ 30fps	760 MB/s	150 MB/s (compressed)
One 128-channel lidar	128 channels @ 10Hz	120 MB/s	40 MB/s
Five millimeter-wave radars	4D imaging radar	300 MB/s	80 MB/s
Total	-	1.18 GB/s	270 MB/s

Memory bottlenecks: NVIDIA Orin-X example

• LPDDR5 theoretical bandwidth: 204 GB/s (dual channel 6400 MT/s)

• Effective bandwidth when multiple sensors are accessed concurrently: < 100 GB/s (< 50%

utilization)

• Reason:

1. Bank conflict: Multiple sensor data streams access the same DRAM Bank at the same time, resulting in queue waiting.

2. Row Buffer Miss: Random access leads to frequent row activation/precharging, each time adding an additional 30–50ns delay.

3. Read and write switching overhead: GPU/NPU switches between reading input data and writing intermediate results, each switching adds 20–30ns delay.

Consequences:

• The computing unit is idling: the time the GPU/NPU waits for data to arrive accounts for 40–60% of the total execution time, and the computing power is "starved".

• Waste of power consumption: Data transfer power consumption accounts for 63.7%, which is much higher than the actual computing power consumption (28.5%).

• Uncontrollable latency: The multi-sensor fusion latency deteriorates from the theoretical 30ms to the measured 80–150ms, which cannot meet the requirements for emergency obstacle avoidance (< 50ms).

1.3 The edge-model impossibility triangle

Autonomous driving is evolving from a modular solution (perception-planning-control) to an end-to-end large model (VLA: Vision-Language-Action). However, end-side deployment faces severe challenges:

Model type	Parameter count	KV-cache requirement	Inference latency (Orin-X)	Power	Problem
Traditional modular design	10B (all models combined)	< 500 MB	50–80ms	60W	Poor generalization and failure on long-tail cases
End-to-end VLA	35–70B	2–5 GB/frame	150–300ms	> 120W	Excessive latency and power consumption

KV-cache bandwidth: a 35B VLA example

• Each frame of inference requires access to KV cache 2–5 GB data (Transformer self-attention mechanism).

• LPDDR5 bandwidth limitation: Even if the theoretical bandwidth is 204 GB/s, the actual available bandwidth is < 100 GB/s, and accessing 3 GB data takes > 30ms.

• Autoregressive generation: Action instructions need to be generated token by token and cannot be parallelized, resulting in linear accumulation of delays.

• Result: The terminal-side VLA inference delay is > 150ms, which cannot meet the L4 real-time requirements (< 50ms).

2. The iNc technology path

2.1 Core technical principles

Lingcun iNc technology fundamentally eliminates data handling by completing analog domain calculations inside the 3D NAND array:

Architecture comparison:

Architecture type	Data path	Bandwidth bottleneck	Energy efficiency	Latency
Traditional architecture	Storage → DRAM → compute → DRAM → storage	Limited DRAM bandwidth (204 GB/s)	Baseline	Baseline
Near-memory computing	Storage → HBM → compute → HBM → storage	HBM bandwidth (4.9 TB/s)	0.6x	2–3x
iNc in-memory computing	Compute inside storage (no data movement)	No bottleneck (analog current summation)	0.02x	43x

Estimated performance of a 512Gb iNc 3D NAND chip

Task type	Dataset	iNc latency	Traditional solution latency	Speedup	Energy-efficiency improvement
Vector similarity search	VGGFace2 (170,000 faces, 512-dimensional features)	50 μs (single match)	2.35 msCPU	47x	43x
Top-600 pooled results	As above	< 500ms (full-database search)	> 20 s	40x	35x
Data transfer volume	As above	14.3–33.3× reduction (pooled sampling)	Baseline	-	-

Performance data

Accuracy loss: only 0.43% (93.39% vs 93.82% ideal value), and the impact on practical applications is negligible.

2.2 iNc architecture in autonomous driving

Core design concept: iNc does not replace SoC, but serves as a heterogeneous co-processor and collaborates with SoC:

Task type	Compute characteristics	Responsible unit	Reason
Control-flow decisions	Branch-intensive and logically complex	SoC CPU50–100 DMIPS	CPU excels at branch prediction and context switches
Trajectory planning	Small matrix operations and real-time optimization	SoC NPU50–100 TOPS	Requires low-latency deterministic execution
Vector search	Large-scale similarity computation (e.g. semantic maps)	iNc	Data-intensive and bandwidth-sensitive

Task type	Compute characteristics	Responsible unit	Reason
Feature fusion	Multi-camera BEV feature fusion	iNc	Cross-sensor access to many historical frames
KV-cache management	Transformer KV-cache reads/writes	iNc	Frequent memory accesses, bandwidth bottleneck
Weight loading	Loading and reusing 35–70B model parameters	iNc	Avoids DRAM occupation

2.3 Performance metrics from compute-in-storage chip data

Refer to the data of a certain model of domestic memory and computing integrated chip:

Metric	An unspecified domestic model	iNc Auto-L4 (STORLEAD projection from similar architectures)	Basis for improvement
Process node	12nm	12nm (automotive grade)	-
Physical computing capacity	256 TOPS (INT8)	Equivalent 500–800 TOPS (vector search)	Analog-domain iNc avoids digital MAC bottlenecks
Typical power	35W	< 60W	iNc data-movement power < 10% (traditional: 63.7%)
Energy-efficiency ratio	7.3 TOPS/W	10–15 TOPS/W	15.6 TOPS/W estimated by analogy with academic papers
Actual throughput	ResNet-50: 8,700 frames/s (batch=1)	BEVFormer: estimated 150–200 frames/s	iNc reduces memory accesses during feature fusion
Validated task	Point-cloud and BEV networks	As above + large VLA model KV cache	-

iNc Auto-L4 versus two NVIDIA Orin-X chips

Metric	Two Orin-X chips	iNc Auto-L4 + lightweight SoC	Improvement
Peak compute performance	508 TOPS (INT8)	SoC: 50 TOPS + iNc: equivalent 500–800 TOPS	5–10× speedup on specific tasks
Effective utilization	20–40%	70–90%	iNc eliminates the bandwidth bottleneck
System power	120W (two chips)	< 80WSoC 20W + iNc 60W	33% reduction
Share of power used for data movement	63.7%	< 10%	6× reduction
BEV fusion latency	80–120ms (measured, DRAM limited)	< 30ms (local fusion)	3–4× reduction
VLA inference latency	150–300ms	< 80ms (local KV-cache access)	2–4× reduction
BOM cost	> CNY 5,000 (dual chips + 16GB LPDDR5)	< CNY 3,500 (single SoC + iNc UFS)	30–40% reduction

3. Disruptive applications

3.1 Sensor fusion: deterministic real-time operation

3.1.1 Timing disaster of traditional solutions

Typical L4 system (8 cameras + 1 lidar + 5 millimeter wave radar):

[Sensor data arrival timing (actual measurement)]

• Cam1: T0 = 0ms

• Cam2: T0 + 3ms (trigger time difference)

• Cam3: T0 + 6ms

• ...

• LiDAR: T0 + 15ms (point cloud processing delay)

• Radar1: T0 + 20ms

• ...

Data processing process (traditional architecture):

1. Each sensor data → DRAM (latency 5–15ms)

2. Timestamp alignment + feature extraction → NPU (latency 30–50ms)

3. BEV feature fusion → NPU (latency 20–40ms, limited by DRAM bandwidth)

4. Target detection + trajectory prediction (latency 20–30ms)

Total latency: 80–150ms (actual measurement)

Question:

• Timing uncertainty: Sensor data arrival time is not synchronized, and timestamp alignment requires buffering multiple frames, increasing latency.

• Bandwidth contention: Multiple sensors access DRAM at the same time, causing bank conflicts and actual bandwidth utilization < 50%.

• Insufficient real-time performance: total delay > 100ms, unable to meet emergency obstacle avoidance (< 50ms) requirements.

3.1.2 Edge preprocessing and in-memory fusion

Architecture Design

Edge Sensor Module (iNc Sensor Hub, 64–128GB eUFS):

• iNc Sensor Hub built into each camera/radar module

• Completed in place: denoising, feature extraction, target detection (YOLO/Transformer)

• Output: Only the feature vector (512 dimensions) + target frame is transmitted, and the data volume is compressed to 10% of the original

• Latency: < 10ms (8 million pixel image → 512-dimensional features)

Central Computing Platform (iNc Auto-L4, 512GB–1TB UFS):

• Receive feature vectors of each sensor and store them in iNc UFS

• BEV feature fusion: Vector weighted fusion is completed inside the NAND array (analog domain current summation)

• Latency: < 20ms (vs traditional solution 40–60ms)

• Output: fused BEV features → SoC NPU for target detection and trajectory prediction

Performance improvement

Stage	Traditional solution latency	iNc solution latency	Improvement
Sensor preprocessing	5–15ms (raw-data upload)	< 10ms (edge feature extraction)	1.5–3x
Data transfer	1.18 GB/s → DRAM	270 MB/s → central iNc (4× compression)	75% less network bandwidth
BEV feature fusion	40–60ms (DRAM bottleneck)	< 20ms (in-memory fusion)	2–3x
Total system latency	80–150ms	< 50ms	2–3x
Power	150W (30W sensors + 120W central)	< 90W (15W sensors + 75W central)	40% reduction

Validation in real-world scenarios

Based on the test of a certain domestic model on an unmanned car:

• BEV network inference: The frame rate is increased from 8 fps (Orin-X) to 25 fps (H30), and the latency is reduced by 3 times.

• Point cloud processing: PointPillars network processes 128-line lidar, and the latency is reduced from 50ms to 18ms.

3.2 End-to-end VLA models in vehicles

3.2.1 "Triple Dilemma" of VLA Large Model

The Vision-Language-Action (VLA) model is a key technology for L5 autonomous driving, which can generate control instructions directly from sensor input without the need for intermediate modules.

Typical VLA model specifications (35B parameters):

Component	Parameter count	Memory use (FP16)	Inference timing characteristics
Visual encoder	5BViT-H	10 GB	Eight parallel camera streams
Language decoder	25BLLaMA-30B	50 GB	Autoregressive, token-by-token output
Action head	5B	10 GB	Steering angle, accelerator and brake outputs

Component	Parameter count	Memory use (FP16)	Inference timing characteristics
KV cache	-	2–5 GB/frame	All historical KV accessed per frame

Three challenges of conventional Orin-X solutions

1. Parameter loading delay:

• The 35B model weights (70 GB FP16) cannot all be put into LPDDR5 (32 GB) and need to be loaded in batches.

• Loading 70 GB parameters from UFS 4.0 (3.6 GB/s) takes >19 seconds, making real-time inference not feasible.

2. KV cache bandwidth black hole:

• Each frame of inference requires access to 2–5 GB KV cache data.

• The actual bandwidth of LPDDR5 is < 100 GB/s, and accessing 3 GB takes > 30ms.

• 30fps inference requires one frame to be completed every 33ms, and KV access has exhausted the time budget.

3. Autoregressive generation delay:

• Action instructions need to be generated token by token (for example, "turn left 15°, accelerate to 40km/h" requires 20+ Tokens).

• Each token generation requires a complete inference (including KV cache access), and the total delay is > 150ms.

Results: Traditional Orin-X runs a 35B VLA model, with inference latency > 200ms and power consumption > 150W, which cannot meet L4/L5 requirements.

3.2.2 Resident weights and local KV-cache management

Architecture design (iNc Auto-L5, 1–2TB UFS):

[Model deployment method]

• 35–70B model weights (FP16/INT8 mixed precision) resident in iNc UFS (no need to load to DRAM)

• During inference, weights are simulated domain matrix multiplied directly with activation values in the NAND array

• Intermediate activation values (< 2 GB) stored in low-capacity LPDDR5 (8 GB is enough)

[KV cache management]

• KV cache (2–5 GB/frame) stored in iNc UFS

• When the Transformer self-attention is calculated, the Query vector directly performs similarity calculation with the KV vector in the NAND array (simulation domain dot product)

• Avoid loading KV cache into DRAM, eliminating bandwidth bottlenecks

• Latency: < 50ms (vs traditional solution > 150ms)

Performance gains: theoretical and academic estimates

Stage	Traditional solution (Orin-X)	iNc Auto-L5	Improvement
Model loading	19+ seconds (UFS to DRAM)	Zero loading (weights resident in iNc)	Unbounded
KV-cache access	30ms/frame (DRAM bandwidth limited)	< 10ms/frame (in-memory compute)	3x
Autoregressive generation	150–300ms (per token)	< 80ms (parallel optimization + low-latency KV access)	2–4x
Total inference latency	200–350ms	< 100ms	2–3.5x
Power	150W	< 80W	47% reduction
DRAM requirement	32 GB (constrained)	8 GB (ample)	4× reduction

3.2.3 Technical feasibility of full L5 autonomy

Scenario: The vehicle is in an unfamiliar mountainous area that it has never been to before, and encounters long-tail scenarios that do not exist in the training data, such as temporary landslides, wild animal crossings, and gesture command.

Traditional modular solution:

• Need to pre-write rules for each scenario (e.g. "Landslope detected → slow down + detour").

• Long-tail scene coverage rate is < 80%, and automatic driving will be exited directly when encountering unknown scenes.

iNc + VLA plan:

• The 70B VLA model runs in real time on iNc Auto-L5 (latency < 100ms, power consumption < 80W).

• The model has "common sense reasoning" capabilities through large-scale pre-training (such as based on 1 million hours of driving videos).

• When encountering a landslide, the model understands that "the road ahead is impassable" → automatically plans a detour route.

• The generalization ability is improved by 3–5 times, and the long-tail scene coverage rate is >95%.

Business Value:

• Say goodbye to dependence on high-precision maps: The cost of collecting and updating high-precision maps is > 1 billion yuan/year/car company, and the VLA model can reduce costs by 80%.

• Accelerate the implementation of L5: From "need to cover all scenarios" (impossible) to "let the model learn to reason" (feasible).

3.3 Intelligent cockpit: context-aware assistance

3.3.1 Localized deployment of multi-modal large models

Scenario: The user said while driving, "Help me find a nearby restaurant with children's amusement facilities. The environment must be quiet. Make a reservation for 4 people at 7pm."

Traditional Plan:

• Voice is uploaded to the cloud (latency > 2 seconds) → Cloud GPT-4 understands the intent → Calls the map API → Returns the result.

• Problem: Unavailable in tunnel/mountainous areas without network; risk of privacy leakage in sensitive conversations (such as business negotiations).

iNc plan (iNc Auto-L3, 256–512GB UFS):

• The 7B multimodal large model (VLM) is resident in iNc UFS and does not require cloud calls.

• Reasoning process:

1. ASR speech recognition (latency < 100ms)

2. VLM understands the intention: "restaurant" + "children's facilities" + "quiet" + "4 people" + "7pm"

3. Local map library (stored in iNc) vector retrieval: matching "child-friendly + quiet" features

4. Call Meituan API to book

5. TTS voice broadcast: "XX restaurant has been reserved for you. It is 3 kilometers away and is expected to arrive in 10 minutes."

• Total delay: < 3 seconds (unperceived by the user)

• Power consumption: < 2W (vs cloud solution needs to maintain 4G/5G connection, power consumption > 5W)

3.3.2 Driver monitoring and emotion recognition

Scenario: The car camera detects that the driver yawned three times in a row + the blink frequency decreased by 30%.

Traditional Plan:

• The DMS model (500M parameters) runs on the SoC NPU and outputs a fatigue score every 100ms.

• Power consumption: 5–8W (continuous operation)

• False alarm rate: 10–15% (not considering individual differences)

iNc solution (iNc Auto-L3):

• The DMS model is deployed in iNc, with inference latency < 20ms and power consumption < 1.5W.

• Personalized modeling: Establish a personalized fatigue model for each user (for example, "a person's normal blink rate is 15 times/min, and if it is less than 10 times/min, he or she is fatigued").

• Emotion recognition: Identify emotions such as anger, anxiety, happiness, etc. through facial micro-expressions + voice intonation, and adjust the cockpit atmosphere (music, lighting,

fragrance).

• Active intervention: After detecting fatigue, it will immediately play fast-paced music + increase the temperature of the air conditioner + voice prompt "It is recommended to rest in the front service area".

Social Value:

• Fatigue driving accounts for >20% of traffic accidents. iNc-driven DMS can provide 5–10 minutes of warning in advance, saving tens of thousands of lives every year.

3.4 Continual learning throughout vehicle life

3.4.1 Personalized driving style learning

Scenario: Users are used to aggressive driving (accelerate quickly, follow the car in front, change lanes decisively), but the automatic driving strategy of the vehicle when leaving the factory is conservative (slow acceleration, large distance between vehicles), resulting in users taking over frequently.

Traditional Plan:

• Data is uploaded to the cloud for training (traffic cost > 1 yuan/GB, training period > 7 days).

• Privacy risk: Driving behavior data (such as frequent locations, driving style) is collected in the cloud.

iNc solution (iNc Auto-L4, supports FP8/BF16 on-chip training):

• Online learning: The scene (camera image, radar data, steering wheel operation) every time the user takes over is automatically recorded as a negative sample.

• Night training: When the vehicle is charging, iNc performs incremental training on the driving strategy model locally (power consumption < 50W, completed in 3 hours).

• Effective the next day: The updated model is automatically deployed, and the takeover rate is reduced by 80% (from 10 times per 100 kilometers to < 2 times).

Technical implementation (calculated based on academic research):

• iNc supports backpropagation gradient calculation (FP8/BF16 accuracy), with an energy efficiency ratio of 15.6 TOPS/W.

• Training data volume: Each takeover records 10 seconds of scene data (approximately 1 GB), and 1,000 takeovers accumulate 1 TB of data.

• Training time: 1 TB data is used to fine-tune the 7B model on iNc Auto-L4, which takes < 3 hours (vs. 1 day to upload to the cloud + 1 day to train).

Disruptive Value:

• Thousands of people, thousands of faces: Each car’s driving strategy is unique and perfectly adapted to the user’s habits.

• Data privacy: Driving data is not uploaded to the cloud and complies with the Personal Information Protection Act.

• Continuous evolution: The vehicle continues to learn throughout its life cycle (10 years+) and

becomes smarter the more it is used.

3.4.2 Regional adaptation through federated learning

Scenario: The vehicle is transferred from Beijing to Tibet for use, facing plateau hypoxia, complex mountain roads, extreme weather and other scenarios that do not exist in the Beijing training data.

Traditional Plan:

• Car companies need to separately collect data, train models, and push OTA for Tibet (cycle > 6 months).

• Takeover rate soared to 20+ times per 100 kilometers.

iNc scheme (federated learning):

• Workshop collaborative training: When multiple vehicles are driving in Tibet, iNc calculates the model gradient locally.

• Gradient aggregation: Each car uploads gradients (< 10 MB) to the cloud for aggregation instead of raw data (> 1 GB).

• Model update: The optimized global model is generated in the cloud and pushed to all vehicles OTA.

• Time cycle: shortened from 6 months to < 1 month, takeover rate reduced to < 2 times/100 kilometers.

Social Value:

• One model can adapt to any region in the world (left-hand drive/right-hand drive, different traffic regulations, complex road conditions), without the need for separate training for each market.

• Accelerate the implementation of L5: quickly accumulate long-tail scenario data through global fleet collaborative learning.

4. Reconstructing the product

4.1 Central computing and integrated memory-compute architecture

Pain points of traditional architecture:

• The vehicle has 100+ ECUs, which control power, chassis, cockpit, and ADAS respectively. They are independent of each other and difficult to coordinate.

• Harness length > 5 kilometers, weight > 60 kg, cost > 5,000 yuan.

iNc driven new architecture:

[Central computing platform (iNc Auto-L4/L5 + lightweight SoC)]

• 1–2 iNc Auto-L4/L5 (for storage + AI computing power)

• 1–2 lightweight SoCs (responsible for control flow + real-time decision-making, 50–100 TOPS)

• Memory pooling via CXL 3.0 interconnect (bandwidth > 1 TB/s)

• Number of ECUs reduced to < 10 (only safety redundancy and actuator control retained)

[Domain controller architecture]

• Intelligent driving domain: iNc Auto-L4 + SoC (responsible for sensing, planning, and control)

• Cockpit domain: iNc Auto-L3 (responsible for multi-modal interaction, entertainment, DMS)

• Power domain: lightweight MCU (responsible for motor and battery management)

• Chassis domain: lightweight MCU (responsible for braking and steering)

Performance and cost optimization

Metric	Traditional architecture	iNc architecture	Improvement
Number of ECUs	100+	< 10	90% reduction
Wiring length	> 5 km	< 1 km	80% reduction
System power	> 300W	< 200W	33% reduction
BOM cost	> CNY 20,000 (chips + ECUs + wiring)	< CNY 12,000	40% reduction
Software iteration cycle	> 2 years (tightly coupled hardware)	< 6 months (software defined)	4× speedup

Case study: Tesla FSD

• Tesla uses central computing architecture + OTA to shorten the function iteration cycle from 2-3 years for traditional car companies to 3-6 months.

• The iNc solution is expected to reduce costs by another 30% on this basis (reducing DRAM + high-end SoC requirements).

4.2 From vehicle sales to computing as a service

4.2.1 Subscription-based autonomous driving

Traditional Mode:

• Users pay a one-time payment for the autonomous driving hardware cost (> 20,000 yuan) when purchasing a car.

• Car companies have no follow-up revenue and low profit margins (< 10%).

New mode for iNc driver:

• Standard hardware configuration: All models are equipped with iNc Auto-L4 out of the factory (cost < 3,500 yuan).

• Software Subscription:

◦ L2+ function: free (high-speed NOA, automatic parking)

◦ L3 function: 199 yuan/month (urban NOA, point-to-point autonomous driving)

◦ L4 function: 399 yuan/month (completely driverless, limited area)

• Car company revenue: Bicycle life cycle (10 years) subscription revenue > 20,000 yuan, gross profit margin > 80%.

Case:

• Tesla FSD subscription fee is US$199/month, the number of subscribers in 2025 is > 500,000, and the annual revenue is > US$1.2 billion.

4.2.2 Renting unused compute capacity

Scenario: When the vehicle is charging at night (8 hours), the computing power of iNc Auto-L4 (equivalent to 500–800 TOPS) is idle.

Business Model:

• Computing power rental: After authorization from the car owner, iNc computing power is leased to cloud computing platforms (such as rendering farms and AI training).

• Revenue sharing: car owners receive 20–50 yuan in revenue per night (electricity cost < 5 yuan).

• Annual income: 7,000-18,000 yuan/year, which can offset part of the car purchase cost.

Technical feasibility:

• iNc power consumption < 60W (power consumption < 0.5 degrees for 8 hours at night, cost < 0.3 yuan).

• Connect to the cloud via 5G/Wi-Fi, receive rendering/training tasks, and upload the output results.

4.3 Robotaxis and autonomous logistics

4.3.1 Autonomous taxis: reduce travel costs by 70%

Traditional Taxi:

• Driver cost: 150 yuan/day (8 hours)

• Fuel cost: 80 yuan/day (200 kilometers × 0.4 yuan/km)

• Vehicle depreciation: 50 yuan/day

• Total cost: 280 yuan/day, cost per kilometer 1.4 yuan

Autonomous Taxi (iNc Auto-L5):

• Driver cost: 0 yuan (unmanned driving)

• Electricity fee: 40 yuan/day (200 kilometers × 0.2 yuan/km)

• Vehicle depreciation: 50 yuan/day (single vehicle cost < 200,000 yuan, life span 10 years)

• Total cost: 90 yuan/day, cost per kilometer 0.45 yuan

• User payment: 0.8 yuan/km (vs. traditional taxi 2 yuan/km)

Commercial Viability:

• The vehicle operates 24 hours a day (no rest for the driver), the average number of orders received per day is > 50, and the daily revenue is > 600 yuan.

• Gross profit margin: > 75%, break-even period < 18 months (vs. traditional taxi > 5 years).

Market Size:

• The global travel market is > 10 trillion US dollars. When Robotaxi penetration reaches 30%, the market size is > 3 trillion US dollars.

4.3.2 Last-mile autonomous logistics

Traditional Express:

• Courier labor cost: 5–8 yuan/order

• Electric tricycle depreciation + electricity fee: 1 yuan/order

• Total cost: 6–9 yuan/order

Unmanned delivery vehicle (iNc Auto-L4):

• Labor cost: 0 yuan

• Electricity fee + depreciation: < 1 yuan/order (cost of a single vehicle < 100,000 yuan, 200 orders delivered per day)

• Total cost: < 1 yuan/order

Market Size:

• China’s daily express delivery orders are >300 million. When the penetration rate of unmanned delivery reaches 50%, the annual cost savings will be >200 billion yuan.

5. Industry impact and social value

5.1 Reshaping the automotive value chain

Stage	Traditional solution	iNc solution	Impact
Sensor	Lidar: CNY 50,000–100,000 each × 1–2	Reduced to 0–1 (V2X supplement)	50–80% lower cost
Compute chip	Two Orin-X chips > CNY 5,000	One iNc Auto-L4 < CNY 3,000	40–60% lower cost
Storage	16GB LPDDR5 + 256GB UFS > CNY 1,000	512GB iNc UFS < CNY 800 (integrated compute)	20–30% lower cost
Software development	Modular solution, cycle > 2 years	End-to-end model, cycle < 1 year	50% lower R&D cost

5.2 Social value

1. Traffic Safety:

• Human error accounts for 94% of the causes of accidents, and the iNc-driven L4/L5 system reduces the accident rate by 90%.

• Lives saved per year: hundreds of thousands (traffic deaths globally >1.3 million per year).

2. Relieving congestion:

• Intelligent dispatch + vehicle-road collaboration improves urban traffic efficiency by 30–50% and reduces commuting time by 20–30%.

3. Energy Saving and Emission Reduction:

• Electric vehicles + autonomous driving optimization (smooth acceleration, fleet coordination) reduce energy consumption by 20–30%.

• Carbon emission reduction: billion tons/year (calculated based on 1 billion vehicles worldwide).

4. Employment Transformation:

• Jobs reduced: Drivers (>100 million globally)

• New positions: AI trainer, remote safety officer, fleet operation, computing power dispatch (estimated > 50 million).

6. Technology roadmap and deployment

6.1 Technology evolution

Phase	Time	Technical target	Application
First generation	2026–2027	iNc Auto-L3: equivalent 100–300 TOPS search performance	Highway NOA, smart cockpit, DMS
Second generation	2027–2028	iNc Auto-L4: equivalent 500–800 TOPS, 35B VLA support	Urban NOA and Robotaxi pilot operation
Third generation	2028–2030	iNc Auto-L5: equivalent 1,000+ TOPS, 100B+ models and on-chip training	Full autonomy and large-scale Robotaxi

6.2 Progress as of January 2026

• Technical Verification: The iNc Auto-L3 prototype has completed bench testing, and the vector retrieval performance reaches the standard (equivalent to 200 TOPS, power consumption < 30W).

• Application Migration: Chongqing University is migrating mainstream autonomous driving algorithms (BEVFormer, UniAD, PointPillars) to iNc architecture.

• Car company cooperation: Signed technical evaluation agreements with two leading new energy car companies, and plans to complete vehicle installation verification in 2027 H1.

• Reference Case: A certain domestic model has been deployed on an unmanned vehicle to verify the feasibility of the integrated storage and computing architecture.

7. Conclusion: redefining the automobile

Conclusion: redefining the automobile

Lingcun iNc in-memory computing technology fundamentally solves the three major dilemmas of autonomous driving by eliminating data transfer bottlenecks, improving computing power utilization, and reducing system power consumption:

• From "computing power inflation" to "computing power efficiency": Increase the effective computing power utilization from 20–40% to 70–90%, and reduce power consumption by 30–50% under the same performance.

• From "cloud dependence" to "vehicle intelligence": 35–70B VLA large model runs in real time on the vehicle (latency < 100ms, power consumption < 80W), achieving true L5 autonomous driving.

• From "hardened at the factory" to "lifelong evolution": Through continuous learning on the vehicle side + federated learning, the vehicle is continuously optimized throughout its life cycle and becomes smarter the more it is used.

In 2026, the "first year of autonomous driving", Lingcun has completed core technology development and is cooperating in depth with Chongqing University and leading car companies. We sincerely invite industry partners to participate in this century-old transformation of the automotive industry, using in-memory computing technology to make safe, efficient, and intelligent travel benefit everyone, and make the vision of "bringing general intelligence into the physical world" a reality.

Get deposit and redefine the car, starting from the deposit calculation.