Feiteninventarisatie drie sporen naar discrete,berekende netstructuren in LLMs

J.Konstapel,18-9-2026.

TL;DR

- Alle drie de sporen zijn met concrete metingen aangetoond in gepubliceerd onderzoek. Ternaire registers (Spoor 3) staan het verst: BitNet b1.58 2B4T (Microsoft, 2 miljard parameters, 4 biljoen tokens) draait op 0,4 GB niet-embedding-geheugen, 29 ms latentie en 0,028 J per token, tegen 0,258 J voor Llama 3.2 1B.
- Berekende/tokenizer-vrije adressen (Spoor 1) en routing-door-passen (Spoor 2) zijn eveneens bewezen: byte-modellen (ByT5, BLT) verplaatsen de embeddingtabel uit de vocabulaire naar rekenlagen (bij mT5-Base is dat 256 miljoen parameters, ~66% van het totaal); hash-routing (Hash Layers) evenaart geleerde gates zonder één routing-parameter.
- De open vraag is niet óf discreet/berekend werkt, maar bij welke schaal en trainingsduur het gelijkwaardig blijft. Onafhankelijk werk (Ouyang et al., ACL 2025) waarschuwt dat het gat tussen lage-bit en fp16 juist groeit naarmate méér trainingstokens worden gebruikt.

Kernbevindingen

Spoor 3 (ternair) is het meest volwassen en best gemeten. BitNet b1.58 toont bij 2–4 miljard parameters gelijkwaardige kwaliteit met dramatisch lagere kosten. Onafhankelijke replicatie bestaat (Spectra/TriLM, Nolano/Mila; Falcon-Edge, TII). De centrale claim "gelijkwaardig aan fp16 bij voldoende schaal" is echter betwist door Ouyang et al.

Spoor 1 (berekende adressen) werkt technisch, maar verschuift kosten. De embeddingtabel verdwijnt uit de vocabulaire, maar byte-modellen krijgen langere sequenties en meer FLOPs. Nieuwe tokens zonder hertraining opnemen is precies waar byte/hash-modellen sterk in zijn.

Spoor 2 (routing door passen) is een bewezen, parametervrije vervanger. Hash-routing gebruikt nul extra parameters en evenaart geleerde gates op perplexiteit. Expert-choice routing lost load-balancing zonder aux-loss op.

De dwarsvraag heeft concrete antwoorden. Er zijn werken die twee sporen combineren (matmul-vrije LM = ternair + element-wise; BLT = byte-adressen + hash-n-gram-embeddings). Meerdere auteurs benoemen expliciet de convergentie richting discrete, hardware-vriendelijke structuren onder geheugen-/energiedruk. Een gemeten drie-in-één-systeem bestaat nog niet.

Details

Spoor 1 — Uitgerekende/deterministische token-adressen

De kern: de embedding wordt niet als vrije parametertabel geleerd, maar deterministisch uit de tokenidentiteit opgebouwd (hash, byte-waarde, of berekende regel).

Hash embeddings (Svenstrup, Hansen, Winther; NeurIPS 2017). Elke token wordt door meerdere hashfuncties naar een kleine gedeelde codeboek-tabel afgebeeld, gewogen met geleerde importantieparameters. Aantal trainbare parameters is B·d + K·k in plaats van K·d bij een standaardtabel (K = vocab-grootte, d = dimensie, B = buckets, k = hashfuncties). De auteurs stellen letterlijk: "Using a hash embedding typically results in a reduction of parameters of several orders of magnitude." Bij gelijke of betere prestaties over een breed scala aan taken; kan miljoenen tokens aan. Hash embeddings waren zelfs marginaal sneller te trainen dan standaardtabellen bij grote vocabulaires. De importantieparameters blijven geleerd, dus het adres is deels berekend, deels geleerd.

ByT5 (Xue et al., TACL 2022). Werkt direct op UTF-8 bytes; vocabulaire = 256 bytes + 3 speciale tokens = 259. Bij mT5-Base geldt: "the vocabulary and softmax output matrices in the mT5-Base model amount to 256 million parameters, or about 66% of the total parameter count." ByT5 verplaatst die parameters naar de transformerlagen. Volgens Tabel 1 van het ByT5-paper daalt het vocabulaire-aandeel van mT5 naar ByT5 per formaat als volgt: Base 66% → 0,1%, Large 42% → 0,06%, XL 27% → 0,04%, XXL 16% → 0,02%. Kwaliteit: ByT5 verslaat mT5 op ruis-gevoelige en spelling-gevoelige taken; nadeel: langere byte-sequenties → meer FLOPs en tragere inferentie. Nieuwe tokens/talen: geen hertraining van een tokenizer nodig, want er is geen vocabulaire.

CANINE (Clark et al., TACL 2022). Eerste tokenizer-vrije encoder; werkt op Unicode-characters via een hashstrategie (geen vocabulaire), met downsampling via strided convoluties om sequentielengte te beheersen. De directe vergelijking CANINE-S vs. mBERT levert +2,8 F1 op TyDi QA met ~28–30% minder parameters, "despite operating on 4x more sequence positions"; met toegevoegde vocab-vrije character-n-grams loopt het verschil op tot +5,7 F1. Toont dat een berekende (gehashte) adressering de vrije tabel kan vervangen én verslaan.

Charformer (Tay et al., 2021). Gebruikt geleerde gradient-based subword tokenization (GBST): scoort character-n-gram-blokken en maakt een zachte selectie. Combineert efficiëntie van subwords met character-niveau. Overtreft subword-modellen op GLUE, meertalige en ruis-datasets.

MegaByte (Yu et al., 2023). Multiscale byte-transformer die bytes in patches groepeert voor miljoen-byte-sequenties; voorloper van BLT.

Byte Latent Transformer / BLT (Pagnoni et al., Meta; ACL 2025). Eerste FLOP-gecontroleerde schaalstudie van byte-modellen tot 8 miljard parameters en 4 biljoen trainingsbytes. Geen vaste vocabulaire; bytes worden dynamisch tot patches gegroepeerd op basis van entropie van de volgende byte. Resultaat: evenaart Llama 3 in bits-per-byte, met tot 50% minder inferentie-FLOPs door grotere gemiddelde patches (6–8 bytes vs. Llama 3's 4,4). BLT gebruikt hash-gebaseerde n-gram embeddingtabellen als hulpmiddel. Belangrijk voorbehoud (onafhankelijke review, Pith): de 50%-claim telt de per-byte kosten van het entropie-model niet volledig mee; bij herberekening kan het voordeel op het 8B/4T-punt kleiner worden. Bij het "Chinchilla compute-optimale" punt wint standaard-Llama nog; BLT wint pas voorbij een crossover-punt met meer training.

Parameter-efficiënte embeddings (arXiv 2505.02266, "PETE"). Bouwt embeddings uit de statistische structuur van BPE-token-ID's in plaats van een vrije tabel; verkent deterministische opbouw. Nog kleinschalig geëvalueerd (NLI, STS-B).

ALONE (Takase & Kobayashi, 2020). Bouwt elk woord-embedding uit één gedeelde embedding plus een deterministische transformatie; parametermatched vergelijkbare BLEU op WMT-vertaling.

Samenvoegen van vocabulaires: het sterkste argument voor berekende adressen is dat byte-/character-/hashmodellen geen schema-vertaling nodig hebben — er is geen vocabulaire dat moet

worden uitgelijnd. Modellen met verschillende BPE-tokenizers kunnen niet zomaar worden samengevoegd; byte-modellen delen per definitie hetzelfde 256-byte-adresruim.

Spoor 2 — Routing/selectie zonder aangeleerde score-gate

Hash Layers (Roller, Sukhbaatar, Szlam, Weston; NeurIPS 2021). Tokens worden naar vaste expert-modules gerouteerd op basis van een hash van de token-ID — geen extra parameters, geen wijziging van de doelfunctie, geen assignment-algoritme. Vergelijkt gunstig met Switch Transformers en BASE Layers op meerdere datasets. Kernpunt: de "gater" is volledig deterministisch en toch competitief. Beperking (opgemerkt door latere werken): evaluaties waren vooral op pre-training-perplexiteit.

DeepSeekMoE auxiliary-loss-free load balancing (Wang, Gao, Zhao, Sun, Dai; 2024, gebruikt in DeepSeek-V3). Introduceert een per-expert bias-term b_i die vóór de top-k-selectie bij de routing-score wordt opgeteld en dynamisch wordt bijgesteld op basis van recente belasting (update $b_i \leftarrow b_i + u \cdot \text{sign}(1 - f_i)$). De bias wordt alleen voor selectie gebruikt, niet voor de gating-waarde. Gevalideerd tot 3 miljard parameters/200 miljard tokens: betere prestatie én betere load-balancing dan aux-loss. Dit is het model waar Konstapels analyse naar verwijst — de selectie is nog steeds gebaseerd op een geleerde score $s_i = h^T e_i$, maar de balancering is verschoven naar een deterministische, telling-gebaseerde regel. Let op: Sigma-MoE-Tiny (2025) meldt dat bij zeer hoge MoE-sparsiteit deze loss-vrije aanpak juist forse onbalans in de onderste lagen kan veroorzaken.

Expert Choice routing (Zhou et al., NeurIPS 2022). Keert de richting om: experts kiezen tokens in plaats van andersom. Garandeert perfecte load-balancing per constructie. Verslaat Hash Layers op gemiddelde scores en variantie ("hashing based routing performs worse than expert choice"), wat aangeeft dat pure load-balancing alleen niet voldoende is voor specialisatie.

Andere deterministische/parametervrije routers: THOR (Zuo et al., 2021, stochastisch met consistentie-regularisatie), fixed-random router (Chen et al., 2023, competitief met geleerde router), taal-specifieke deterministische routing (Fan et al., 2021). Meerdere werken merken op dat routing snel convergeert naar token-ID-gebaseerde routing in de vroege trainingsfase (Xue et al., 2024), wat suggereert dat de geleerde router weinig toevoegt.

Weggevallen parameters: hash-routing verwijdert de volledige router-matrix (bij DeepSeek-schaal is dat $N_e \times d_{model}$ per MoE-laag). De prijs is minder expert-specialisatie dan met een geleerde of expert-choice router.

Spoor 3 — Ternaire registers

Oudere basis. Ternary Weight Networks (Li & Liu, 2016) en Trained Ternary Quantization (Zhu, Han, Mao, Dally; ICLR 2017) beperken gewichten tot $\{-1,0,+1\}$ met een schaalfactor. TTQ: ~16× kleiner dan fp32; op ImageNet/AlexNet 0,3% Top-1 beter dan fp én 3% beter dan eerdere ternaire modellen; op CIFAR-10 ResNet-32/44/56 zelfs 0,04–0,36% beter dan fp. Dit vestigde al vroeg dat ternair (2-bit opslag) met minimale of geen kwaliteitsverlies kan.

BitNet b1.58 (Ma et al., Microsoft; arXiv 2402.17764, feb 2024). Elk gewicht ternair $\{-1,0,+1\}$ = $\log_2(3) \approx 1{,}58$ bit. Getraind vanaf nul met BitLinear (vervangt nn.Linear). Claim: evenaart fp16/bf16 bij gelijke modelgrootte en trainingstokens in perplexiteit én eindtaak, vanaf 3B. Gemeten bij 3B/2T tokens: BitNet b1.58 3B verslaat StableLM-3B (gemiddelde 74,34 vs 73,22). Bij 3B is BitNet 3,55× minder GPU-geheugen en 2,71× sneller dan fp16-Llama. Gestelde equivalenties (afgeleid uit hun schaalcurves): 13B BitNet efficiënter dan 3B fp16; 30B BitNet efficiënter dan 7B fp16; 70B BitNet efficiënter dan 13B fp16.

BitNet b1.58 2B4T (Ma et al., Microsoft; arXiv 2504.12285, apr 2025). Eerste open-source native 1-bit LLM op 2B-schaal, 4 biljoen tokens, W1.58A8 (ternaire gewichten, 8-bit activaties), LLaMA-3-tokenizer (vocab 128.256). Gemeten efficiëntie (niet-embedding geheugen; CPU-latentie per token; geschatte energie):

- BitNet b1.58 2B: **0,4 GB / 29 ms / 0,028 J**
- Llama 3.2 1B: 2 GB / 48 ms / 0,258 J
- Gemma-3 1B: 1,4 GB / 41 ms / 0,186 J
- Qwen2.5 1.5B: 2,6 GB / 65 ms / 0,347 J
- MiniCPM 2B: 4,8 GB / 124 ms / 0,649 J

Kwaliteit (instruction-tuned): GSM8K 58,38 (hoogste in klasse, vs Llama 38,21, Qwen 56,79); WinoGrande 71,90; PIQA 77,09; gemiddelde 54,19. Belangrijke nuance: op het 11-benchmark-gemiddelde ligt BitNet (54,19) iéts onder Qwen2.5-1.5B (55,23), maar ruim boven Llama 3.2 1B (44,90). Qwen wint op MMLU (60,25 vs 53,17), MATH-500 (53,00 vs 43,40) en HumanEval+ (50,60 vs 38,40); BitNet wint op GSM8K, ARC-Challenge, WinoGrande en PIQA. Tegen INT4-PTQ van Qwen2.5-1.5B houdt BitNet een sterker gemiddelde (55,01 vs 52,15 GPTQ) met minder geheugen (0,4 vs 0,7 GB).

BitNet a4.8 (Wang, Ma, Wei; arXiv 2411.04965). Voegt 4-bit activaties toe (gewichten blijven ternair). Hybride kwantisatie + sparsificatie tegen outliers. Prestatie vergelijkbaar met b1.58, sneller door INT4/FP4-kernels; activeert slechts 55% van de parameters; ondersteunt 3-bit KV-cache.

Onafhankelijke replicatie op schaal — Spectra/TriLM (Kaushal, Vaidhya et al., Nolano AI/ Mila/UdeM; ICLR 2025, arXiv 2407.12327). Eerste open suite die FloatLM, QuantLM en TriLM over 99M–3,9B naast elkaar traint (300B tokens, 54 modellen, 500+ checkpoints). Kernresultaat: de 3,9B TriLM evenaart de FloatLM 3,9B op álle benchmarks, terwijl hij minder bits heeft dan een FloatLM 830M. TriLM's overtreffen QuantLM's; TriLM 3,9B verslaat QuantLM-3bit 3,9B. Spectra 1.1 (arXiv 2506.23025) traint tot 1,2 biljoen tokens en vindt dat TriLM's méér profiteren van extra data dan van extra parameters; introduceert 2-bit en 1,6-bit packing.

Onafhankelijke replicatie — Falcon-Edge/Falcon-E (TII, mei 2025). 1B en 3B ternaire modellen op BitNet-basis, base + instruct. Levert in één training een bf16-, een native BitNet- en een pre-gekwantiseerde variant. Toolkit `onebitllms` open-source. Bevestigt dat de BitNet-recipe reproduceerbaar is buiten Microsoft.

Matmul-vrije LM (Zhu et al., NeurIPS 2024; arXiv 2406.02528). Combineert ternaire gewichten (dense lagen → optellen) met element-wise Hadamard/GRU voor self-attention → elimineert álle matrixvermenigvuldigingen. Getest tot 2,7B; gat met fp krimpt bij grotere schaal. GPU-implementatie: tot 61% minder geheugen bij training, 10× minder bij inferentie, 4,57× snellere inferentie. 13B matmul-vrij model: 4,19 GB GPU-geheugen (695 ms latentie) vs 48,5 GB (3183 ms) voor de equivalente transformer. FPGA-implementatie: miljard-parameter-schaal op 13 W. Op neuromorfe hardware (Loihi 2): claim van hogere doorvoer bij lagere energie.

Inferentie-implementaties en hardware.

- **bitnet.cpp (Wang et al., ACL 2025; arXiv 2502.11880).** mpGEMM-kernels (TL, I2_S) voor ternaire LLM's op edge. Tot 6,25× sneller dan fp-baseline en tot 2,32× sneller dan lage-bit-baselines. Verliesloze inferentie voor BitNet b1.58; haalt 7,45 tokens/s op Apple M2 Ultra bij een 100B ternair model.
- **llama.cpp TQ-formaten.** TQ1_0 = 1,6875 bit-per-gewicht (pakt 5 ternaire waarden in 1 byte, want 3^5=243<256); TQ2_0 = 2,0625 bit (sneller, meer ruimte). Recent werk (CAT-Q, TWLA, PT²-LLM) meldt met deze formaten 1,3–3,6× snelheidswinst en >80%

geheugenbesparing t.o.v. fp16 (bijv. TWLA op LLaMA2-13B: 3,64× sneller dan fp16, geheugen van 23,7 GB naar 3,34 GB).

- **Balanced ternary in hardware.** De Setun (Moskou, 1958) gebruikte balanced ternary {−1,0,+1}; er werden vijftig machines gebouwd van 1959 tot 1965, ontwikkeld o.l.v. Sergei Sobolev en Nikolay Brusentsov en gefabriceerd in de Kazan Mathematical plant. Historisch commentaar (Hayes): het ternaire voordeel werd "verspild" omdat elk trit in twee magnetische kernen werd opgeslagen. Knuth noemde ternaire rekenkunde "perhaps the prettiest number system of all". Recent: CNTFET- en single-electron-transistor-implementaties van ternaire logica; theoretisch is drie de optimale radix qua "cost of complexity". Recente ternaire accelerators (BitROM, T-SAR, LUT-gebaseerde accelerators) mikken specifiek op 1,58-bit inferentie.

De centrale schaal-claim en de tegenspraak. De BitNet-claim is dat het gat met fp16 krimpt naarmate het model groeit. Spectra (3,9B TriLM = 3,9B FloatLM) ondersteunt dit onafhankelijk. Maar Ouyang et al. ("Low-Bit Quantization Favors Undertrained LLMs", ACL 2025; arXiv 2411.17691) leggen de andere dimensie bloot: het gat groeit met méér trainingstokens. Hun kwantisatie-geïnduceerde degradatie (QiD) volgt $\Delta qLoss = k \cdot D^{\beta}/(N^{\alpha} \cdot P^{\gamma})$, gefit als $0{,}017 \cdot D^{0{,}525}/(N^{0{,}226} \cdot P^{5{,}50})$, met D = tokens ($\beta \approx 0{,}53$, in de teller → degradatie stíjgt met data), N = parameters, P = bits. Basis: 1500+ Pythia-checkpoints (160M–12B, 1B–206B tokens; 2/3/4-bit via GPTQ, met AWQ- en bitsandbytes-ablaties). Projectie: een 70B-model heeft >17 biljoen tokens nodig voordat 4-bit-kwantisatie een QiD>0,2 oploopt; een 405B-model bijna 50 biljoen. Bij 100 biljoen tokens (verwacht 2025–2026) wordt lage-bit "may NOT be desirable", vooral voor kleinere modellen.

Cruciale voorbehouden bij die tegenwerping:

1. De kern-scalingwet is afgeleid van **post-training-kwantisatie (PTQ)**, niet van QAT. De 100T-cijfers zijn **extrapolaties**; gemeten is tot 300B tokens (Pythia).
2. Voor native ternair (BitNet, QAT) tonen ze een **kleinschalige replicatie** (120M en 1,2B): de BitNet-verliescurve loopt aanvankelijk gelijk met bf16, maar begint later achter te lopen — "the gap manifests later compared to post-quantization". Ze merken op dat de originele BitNet-cijfers tot ~100 miljard tokens gingen en "express concerns about their performance at larger training scales". BitNet 2B4T (4T tokens) is groter dan hun replicatie, maar nog ver onder het 100T-regime; en het feit dat 2B4T iets onder Qwen2.5-1.5B scoort op het gemiddelde is consistent met (niet bewijzend voor) de these van Ouyang.

Dwarsvraag — combinaties en benoemde convergentie

Combinaties van twee of meer sporen:

- **Matmul-vrije LM = ternair (Spoor 3) + deterministische element-wise operaties.** Vervangt zowel de vermenigvuldiging als de dichte-laag-structuur.
- **BLT = berekende byte-adressen (Spoor 1) + hash-n-gram embeddings.** Gebruikt hashing zowel voor adressering als voor hulp-embeddings.
- **DeepSeek-V3 zelf = top-k selectie + loss-vrije (telling-gebaseerde) balancering (Spoor 2) + lagerang-KV (MLA) + FP8.** Precies het model uit Konstapels analyse; het benadert al drie van zijn vier netoperaties, maar houdt geleerde vocab-embeddings en continue registers vast.
- In het beschikbare onderzoek vond ik **geen enkel werk dat alle drie de sporen tegelijk combineert** (ternair + hash-routing + berekende adressen in één model). Dat is een reële witte vlek. Een ternaire MoE met hash-routing en byte-adressen bestaat, voor zover deze inventarisatie reikt, nog niet als gemeten systeem.

Wie benoemt de convergentie expliciet? Meerdere bronnen benoemen de beweging richting discrete, deterministische, hardware-vriendelijke structuren onder energie-/geheugendruk:

- BitNet b1.58 zelf: "defines a new scaling law and recipe" en "enables a new computation paradigm and opens the door for designing specific hardware optimized for 1-bit LLMs".
- Emergent Mind (samenvattend): "future hardware-software co-designs will increasingly target ternary/integer arithmetic as a fundamental substrate".
- Matmul-vrije LM: expliciete "call to action" om lichtgewicht, hardware-vriendelijke architecturen te prioriteren.
- Spectra: motiveert ternair vanuit de "memory wall" — GPU-rekenkracht groeit sneller dan geheugenbandbreedte, wat lage-bit onvermijdelijk maakt.
- Byte/tokenizer-vrije lijn (ByT5, CANINE, BLT): expliciet framing dat vaste vocabulaires "a model's ability to adapt" beperken en dat berekende adressering universeler en robuuster is.

De convergentie wordt dus per spoor benoemd, maar niemand formuleert de vier-operatie-eenheid zoals Konstapel; de veldbeweging naar discreet + berekend + deterministisch is wél expliciet erkend, gedreven door energie en geheugen, niet door theorie.

Aanbevelingen

Wat de tritvariant nog moet aantonen dat níét al is aangetoond — gefaseerd:

Fase 1 — Claim scherpstellen tegen wat al bewezen is. Ternaire gewichten bij 2–4B (BitNet, Spectra, Falcon-E) zijn géén open vraag meer; die parity is gemeten. Bouw de claim dus niet op "ternair werkt", maar op de combinatie. Drempel: als een nieuw voorstel alleen ternaire gewichten test, voegt het niets toe aan de literatuur.

Fase 2 — De onbewezen combinatie aanvallen. De echte witte vlek is één model dat berekende adressen (Spoor 1) + routing-door-passen (Spoor 2) + ternaire registers (Spoor 3) tegelijk draagt, met een "zelfde doorgang, minder"-meting. Concreet: een byte- of hash-geadresseerde, hash-gerouteerde, ternaire MoE. Benchmark tegen een fp16/bf16 model met geleerde embeddings en geleerde gate op gelijke kwaliteit (perplexiteit + eindtaken), en rapporteer parameters, geheugen, energie (J/token) en latentie.

Fase 3 — De schaal-token-dimensie meten, niet alleen de parameter-dimensie. Ouyang et al. is de sterkste tegenwerping: het gat groeit met trainingstokens. Elke serieuze tritclaim moet de kwaliteit meten bij tenminste 1–4 biljoen tokens, niet alleen bij een vast budget. Drempel die de conclusie zou omdraaien: als het ternaire model bij >4T tokens nog steeds binnen ~1 punt van fp16 blijft op het gemiddelde, ondermijnt dat de these van Ouyang; als het gat groeit zoals ~D^0,53, bevestigt het die.

Fase 4 — Hardware-realisatie apart bewijzen. De theoretische ternaire winst wordt op GPU's deels tenietgedaan omdat die dequantiseren naar hogere precisie. De echte energiewinst is aangetoond op bitnet.cpp (CPU, 6,25×), FPGA (13 W) en neuromorfe chips (Loihi 2). Meet op zulke hardware, niet op een standaard-GPU, anders is de energieclaim niet overtuigend.

Wat je NIET opnieuw hoeft te bewijzen: ternaire parity bij ~3B (bewezen), hash-routing evenaart geleerde gates op perplexiteit (bewezen), embeddingtabel is vervangbaar door berekend adres (bewezen). Focus alle nieuwe metingen op de combinatie en op het gedrag bij grote trainingsbudgetten.

Caveats

- **"Gelijkwaardig" is contextafhankelijk.** BitNet 2B4T evenaart fp16-peers op sommige taken (GSM8K, WinoGrande) maar ligt iets onder de sterkste peer (Qwen2.5-1.5B) op het gemiddelde en duidelijk onder op MMLU en code. "Zelfde doorgang" geldt dus taak-afhankelijk, niet universeel.
- **De schaal-claim is nog niet definitief beslecht.** BitNet en Spectra ondersteunen parity-bij-schaal langs de parameter-as; Ouyang et al. betwisten het langs de token-as. Beide gebruiken deels extrapolatie (Ouyang's 100T-cijfers zijn projecties op basis van metingen tot 300B; BitNet's grote-schaal-equivalenties zijn afgeleid, niet alle direct gemeten).
- **Trainingskosten blijven fp.** Ternaire QAT houdt "shadow weights" in bf16 tijdens training aan; de winst zit vrijwel volledig in inferentie, niet in training.
- **Byte-modellen verschuiven kosten.** Embeddingsparameters verdwijnen, maar sequenties worden langer en FLOPs stijgen; de BLT-inferentiewinst is bovendien door onafhankelijke review genuanceerd omdat de entropie-model-kosten niet volledig zijn meegerekend.
- **Hash-routing offert specialisatie.** Parametervrij en load-gebalanceerd, maar expert-choice routing verslaat het op kwaliteit — pure load-balancing genereert niet vanzelf gespecialiseerde experts.
- **Geen gemeten drie-in-één.** De combinatie van alle drie de sporen in één gemeten systeem bestaat nog niet in de gevonden literatuur; dat is zowel het risico als de kans voor de tritvariant.
- **Geen bron formuleert Konstapels vier-operatie-eenheid.** De convergentie naar discreet/berekend/deterministisch wordt per spoor erkend en aan energie/geheugen toegeschreven, maar niet als één theoretisch geheel.

Geannoteerde referentielijst

1. **Ma et al., "The Era of 1-bit LLMs: All LLMs are in 1.58 Bits" (arXiv 2402.17764, 2024).** De oorsprong van de 1,58-bit-claim. *Waarom lezen?* Bevat de kern-scaling-claim en de equivalentietabel (13B trit ≈ 3B fp16). *Leesadvies:* kort; lees samen met het 2B4T-rapport voor de gemeten cijfers.

2. **Ma et al., "BitNet b1.58 2B4T Technical Report" (arXiv 2504.12285, 2025).** De best gemeten ternaire LLM. *Waarom lezen?* Tabel 1 en 2 geven de harde geheugen-/energie-/latentie- en benchmarkcijfers. *Leesadvies:* begin hier voor Spoor 3; let op de nuance dat het gemiddelde iets onder Qwen ligt.

3. **Kaushal, Vaidhya et al., "Spectra: Pretraining Ternary Language Models at Scale" (ICLR 2025, arXiv 2407.12327) + Spectra 1.1 (arXiv 2506.23025).** Onafhankelijke replicatie buiten Microsoft. *Waarom lezen?* Sterkste onafhankelijke bewijs voor parity-bij-schaal (3,9B TriLM = 3,9B FloatLM) en de token-vs-parameter-bevinding. *Leesadvies:* essentieel als tegenwicht/steun voor de BitNet-claim.

4. **Ouyang et al., "Low-Bit Quantization Favors Undertrained LLMs" (ACL 2025, arXiv 2411.17691).** De belangrijkste tegenwerping. *Waarom lezen?* Formuleert de scalingwet waarin het lage-bit-gat groeit met trainingstokens; bevat een directe BitNet-replicatie en expliciete "concerns". *Leesadvies:* lees §4.4 en voetnoot 6; onthoud dat de kernwet PTQ betreft, de BitNet-kritiek een kleinschalige hypothese is.

5. **Zhu, Han, Mao, Dally, "Trained Ternary Quantization" (ICLR 2017, arXiv 1612.01064).** De klassieke ternaire basis. *Waarom lezen?* Toont al in 2017 ~16× compressie met gelijke of betere accuracy. *Leesadvies:* historische context; kort.

6. **Zhu et al., "Scalable MatMul-free Language Modeling" (NeurIPS 2024, arXiv 2406.02528).** Combineert ternair + deterministische element-wise ops. *Waarom lezen?*

Concrete geheugen-/energiecijfers (61% training, 10× inferentie, 13B in 4,19 GB) én FPGA/neuromorfe metingen. *Leesadvies:* het dichtst bij een multi-spoor-systeem; lees voor de hardware-argumenten.

7. **Wang et al., "bitnet.cpp: Efficient Edge Inference for Ternary LLMs" (ACL 2025, arXiv 2502.11880).** De inferentie-engine. *Waarom lezen?* Bewijst dat de ternaire winst op CPU realiseerbaar is (6,25×), niet slechts theoretisch. *Leesadvies:* lees als je de energieclaim wilt hardmaken op echte hardware.

8. **Roller et al., "Hash Layers for Large Sparse Models" (NeurIPS 2021).** Fundament van Spoor 2. *Waarom lezen?* Parametervrije, deterministische routing die geleerde gates evenaart. *Leesadvies:* kern voor "routing door passen"; let op dat evaluatie vooral perplexiteit betreft.

9. **Zhou et al., "Mixture-of-Experts with Expert Choice Routing" (NeurIPS 2022).** De sterkere tegenhanger van hash-routing. *Waarom lezen?* Toont dat load-balancing alleen niet volstaat; expert-choice verslaat hashing op kwaliteit. *Leesadvies:* lees naast Hash Layers voor de kwaliteit-vs-parametervrijheid-afweging.

10. **Wang, Gao, Zhao, Sun, Dai, "Auxiliary-Loss-Free Load Balancing" (2024, arXiv 2408.15664).** De DeepSeek-V3-routingmethode. *Waarom lezen?* Precies het mechanisme uit Konstapels analyse; deterministische, telling-gebaseerde balancering. *Leesadvies:* kort; lees met het DeepSeek-V3-rapport (arXiv 2412.19437).

11. **Xue et al., "ByT5: Towards a Token-Free Future" (TACL 2022, arXiv 2105.13626).** Fundament van Spoor 1. *Waarom lezen?* Kwantificeert de ~66%-embeddingparameters bij mT5-Base en de verschuiving naar rekenlagen (Tabel 1). *Leesadvies:* kern voor "berekende adressen"; let op de FLOPs-trade-off.

12. **Pagnoni et al., "Byte Latent Transformer: Patches Scale Better Than Tokens" (ACL 2025, arXiv 2412.09871).** De meest recente byte-doorbraak. *Waarom lezen?* FLOP-gecontroleerde schaalstudie tot 8B; combineert byte-adressen met hash-embeddings. *Leesadvies:* lees met de Pith-review over de genuanceerde 50%-claim.

13. **Clark et al., "CANINE" (TACL 2022, arXiv 2103.06874).** Eerste tokenizer-vrije encoder via hashing. *Waarom lezen?* Bewijst dat gehashte (berekende) adressering een vrije vocabulaire kan vervangen én verslaan (+2,8 F1 CANINE-S met ~28–30% minder parameters; tot +5,7 F1 met n-grams). *Leesadvies:* kort; complementair aan ByT5.

14. **Svenstrup, Hansen, Winther, "Hash Embeddings" (NeurIPS 2017, arXiv 1709.03933).** De hash-embedding-basis. *Waarom lezen?* Parameterreductie van "several orders of magnitude" bij gelijke prestatie. *Leesadvies:* lees voor de wiskunde van berekende adressering ($B \cdot d + K \cdot k$).

15. **Wang, Ma, Wei, "BitNet a4.8" (arXiv 2411.04965).** De opvolger met 4-bit activaties. *Waarom lezen?* Toont dat ook activaties omlaag kunnen (55% actieve parameters, 3-bit KV). *Leesadvies:* kort; relevant als je "minder" ook op activaties wilt betrekken.