DeepSeek-V4 Training

Deep Dive into DeepSeek — Chapter 13

Yin Yang

Foundation Books Project

The one rule of this chapter

Every transition must name the state being changed, its owner, and the representation handed to the next stage.

- V4's hybrid attention and mHC block define a forward path. That path becomes a released policy only when curricula, distributed layouts, optimizer owners, and precision paths agree with it.
- Pretraining runs 32–33 trillion tokens through a dense-to-sparse attention curriculum, compressed context-parallel exchange, routing controls, and disjoint Muon/AdamW ownership.
- Post-training then quantizes selected operators, cultivates domain specialists, and consolidates them into one learner.

Goal: follow the training-state ledger from the dense foundation through policy consolidation, keeping source-reported claims, inspected configuration, and the book's own derived traces on their proper side of the claim boundary.

Roadmap

1. Pretraining scale and the attention curriculum

Roadmap

1. Pretraining scale and the attention curriculum
2. Compressed-attention parallelism and routing controls

Roadmap

1. Pretraining scale and the attention curriculum
2. Compressed-attention parallelism and routing controls
3. Muon: ownership and update geometry

Roadmap

1. Pretraining scale and the attention curriculum
2. Compressed-attention parallelism and routing controls
3. Muon: ownership and update geometry
4. Low-precision QAT

Roadmap

1. Pretraining scale and the attention curriculum
2. Compressed-attention parallelism and routing controls
3. Muon: ownership and update geometry
4. Low-precision QAT
5. Post-training: specialists, GRM, and on-policy distillation

Roadmap

1. Pretraining scale and the attention curriculum
2. Compressed-attention parallelism and routing controls
3. Muon: ownership and update geometry
4. Low-precision QAT
5. Post-training: specialists, GRM, and on-policy distillation
6. Evaluation surfaces and open questions

Roadmap

1. Pretraining scale and the attention curriculum
2. Compressed-attention parallelism and routing controls
3. Muon: ownership and update geometry
4. Low-precision QAT
5. Post-training: specialists, GRM, and on-policy distillation
6. Evaluation surfaces and open questions
7. Summary and what carries forward

Pretraining scale and the attention curriculum

Two runs, one outer recipe

Run field	V4-Flash	V4-Pro
Consumed tokens	32T	33T
Max batch	75.5M tokens	94.4M tokens
Peak / endpoint LR	2.7×10^{-4} / 2.7×10^{-5}	2.0×10^{-4} / 2.0×10^{-5}
MTP loss weight	0.3 for most of training, then 0.1 at LR decay	

[REPORTED: DeepSeek-V4 technical report]

Corpus scale and schedule shape are reported; mixture weights, deduplication state, and exact token order stay private.

Sparse attention is trained in, not switched on

Stage	Attention / length	What changes
Dense foundation	Dense; 4K	Flash: 1T tokens; Pro: a longer, unquantified dense stage.

[REPORTED: DeepSeek-V4 technical report]

Sparse attention is trained in, not switched on

Stage	Attention / length	What changes
Dense foundation	Dense; $4\mathrm{K}$	Flash: 1T tokens; Pro: a longer, unquantified dense stage.
Length extension	Dense; $16\mathrm{K} \rightarrow 64\mathrm{K}$	Longer sequences, dense target path retained.

[REPORTED: DeepSeek-V4 technical report]

Sparse attention is trained in, not switched on

Stage	Attention / length	What changes
Dense foundation	Dense; $4\mathrm{K}$	Flash: 1T tokens; Pro: a longer, unquantified dense stage.
Length extension	Dense; $16\mathrm{K} \rightarrow 64\mathrm{K}$	Longer sequences, dense target path retained.
Indexer warm-up	Dense target + CSA indexer; $64\mathrm{K}$	The lightning indexer learns what dense attention would have read.

[REPORTED: DeepSeek-V4 technical report]

Sparse attention is trained in, not switched on

Stage	Attention / length	What changes
Dense foundation	Dense; $4\mathrm{K}$	Flash: 1T tokens; Pro: a longer, unquantified dense stage.
Length extension	Dense; $16\mathrm{K} \rightarrow 64\mathrm{K}$	Longer sequences, dense target path retained.
Indexer warm-up	Dense target + CSA indexer; $64\mathrm{K}$	The lightning indexer learns what dense attention would have read.
Sparse continuation	CSA/HCA; up to $1\mathrm{M}$	The trained selector joins the active path; the full model adapts to it.

[REPORTED: DeepSeek-V4 technical report]

Sparse attention is trained in, not switched on

Stage	Attention / length	What changes
Dense foundation	Dense; $4\mathrm{K}$	Flash: 1T tokens; Pro: a longer, unquantified dense stage.
Length extension	Dense; $16\mathrm{K} \rightarrow 64\mathrm{K}$	Longer sequences, dense target path retained.
Indexer warm-up	Dense target + CSA indexer; $64\mathrm{K}$	The lightning indexer learns what dense attention would have read.
Sparse continuation	CSA/HCA; up to $1\mathrm{M}$	The trained selector joins the active path; the full model adapts to it.

[REPORTED: DeepSeek-V4 technical report]

The selector is trained against the dense path *before* it is allowed to gate what the model actually attends to.

Compressed-attention parallelism and routing controls

Exchange the overlap, then exchange the summaries

Definition (V4 compressed-attention CP exchange)
A context-parallel group of p ranks each owns s uncompressed KV entries at compression rate r. Boundary stage: rank i sends its last r entries to rank $i+1$ so a crossing compression group can be finished. Global stage: ranks all-gather the resulting candidates, and select-and-pad builds a fixed $p(s/r)$-slot global compressed layout.

Exchange the overlap, then exchange the summaries

Definition (V4 compressed-attention CP exchange)
A context-parallel group of p ranks each owns s uncompressed KV entries at compression rate r. Boundary stage: rank i sends its last r entries to rank $i+1$ so a crossing compression group can be finished. Global stage: ranks all-gather the resulting candidates, and select-and-pad builds a fixed $p(s/r)$-slot global compressed layout.

For $p = 2$, $s = 8$, $m = 4$: rank 0 sends positions 5–8 once; each rank emits $s/m + 1 = 3$ candidates; six are all-gathered; select-and-pad builds $ps/m = 4$ global slots.

Exchange the overlap, then exchange the summaries

Definition (V4 compressed-attention CP exchange)
A context-parallel group of p ranks each owns s uncompressed KV entries at compression rate r. Boundary stage: rank i sends its last r entries to rank $i+1$ so a crossing compression group can be finished. Global stage: ranks all-gather the resulting candidates, and select-and-pad builds a fixed $p(s/r)$-slot global compressed layout.

For $p = 2$, $s = 8$, $m = 4$: rank 0 sends positions 5–8 once; each rank emits $s/m + 1 = 3$ candidates; six are all-gathered; select-and-pad builds $ps/m = 4$ global slots.

No uncompressed KV block makes a full circuit of the ranks — unlike Ring Attention.

[REPORTED: DeepSeek-V4 technical report; cf. Ring Attention, arXiv:2310.01889]

Two controls after a routing decision — and two for stability

$$\widetilde{s}_{i,t} = s_{i,t} + b_i \quad \text{selects } \mathcal{T}_t, \qquad g_{i,t} = \rho \frac{s_{i,t}}{\sum_{j \in \mathcal{T}_t} s_{j,t}} \quad \text{combines } i \in \mathcal{T}_t.$$

- The correction bias b_i changes *membership*; the uncorrected affinity $s_{i,t}$ still sets the *mixture weight*.

[REPORTED: DeepSeek-V4 technical report]

Two controls after a routing decision — and two for stability

$$\widetilde{s}_{i,t} = s_{i,t} + b_i \quad \text{selects } \mathcal{T}_t, \qquad g_{i,t} = \rho \frac{s_{i,t}}{\sum_{j \in \mathcal{T}_t} s_{j,t}} \quad \text{combines } i \in \mathcal{T}_t.$$

- The correction bias b_i changes *membership*; the uncorrected affinity $s_{i,t}$ still sets the *mixture weight*.
- Anticipatory routing recomputes expert indices from stale parameters $\theta_{u-\Delta u}$ after a detected spike; SwiGLU's linear branch is clamped to $[-10, 10]$ throughout training.

[REPORTED: DeepSeek-V4 technical report]

Two controls after a routing decision — and two for stability

$$\widetilde{s}_{i,t} = s_{i,t} + b_i \quad \text{selects } \mathcal{T}_t, \qquad g_{i,t} = \rho \frac{s_{i,t}}{\sum_{j \in \mathcal{T}_t} s_{j,t}} \quad \text{combines } i \in \mathcal{T}_t.$$

- The correction bias b_i changes *membership*; the uncorrected affinity $s_{i,t}$ still sets the *mixture weight*.
- Anticipatory routing recomputes expert indices from stale parameters $\theta_{u-\Delta u}$ after a detected spike; SwiGLU's linear branch is clamped to $[-10, 10]$ throughout training.

[REPORTED: DeepSeek-V4 technical report]

Reusing the corrected score as an output weight would make the MoE function change every time the load-balancing bias moves.

Muon: ownership and update geometry

Disjoint ownership, one update per parameter

$$\Theta_{\mathrm{A}} = \Theta_{\mathrm{embed}} \cup \Theta_{\mathrm{head}} \cup \Theta_{\mathrm{mHC\ bias}} \cup \Theta_{\mathrm{mHC\ gate}} \cup \Theta_{\mathrm{RMSNorm}}, \qquad \Theta_{\mathrm{M}} = \Theta \setminus \Theta_{\mathrm{A}}, \qquad \Theta_{\mathrm{A}} \cap \Theta_{\mathrm{M}} = \varnothing.$$

- AdamW owns embeddings, head, mHC biases/gates, RMSNorm weights.

[REPORTED: DeepSeek-V4 technical report]

Disjoint ownership, one update per parameter

$$\Theta_{\mathrm{A}} = \Theta_{\mathrm{embed}} \cup \Theta_{\mathrm{head}} \cup \Theta_{\mathrm{mHC\ bias}} \cup \Theta_{\mathrm{mHC\ gate}} \cup \Theta_{\mathrm{RMSNorm}}, \qquad \Theta_{\mathrm{M}} = \Theta \setminus \Theta_{\mathrm{A}}, \qquad \Theta_{\mathrm{A}} \cap \Theta_{\mathrm{M}} = \varnothing.$$

- AdamW owns embeddings, head, mHC biases/gates, RMSNorm weights.
- Every other matrix is Muon-owned: momentum, Nesterov look-ahead, hybrid Newton–Schulz, shape-aware rescale, decay, subtract.

[REPORTED: DeepSeek-V4 technical report]

Disjoint ownership, one update per parameter

$$\Theta_{\mathrm{A}} = \Theta_{\mathrm{embed}} \cup \Theta_{\mathrm{head}} \cup \Theta_{\mathrm{mHC\ bias}} \cup \Theta_{\mathrm{mHC\ gate}} \cup \Theta_{\mathrm{RMSNorm}}, \qquad \Theta_{\mathrm{M}} = \Theta \setminus \Theta_{\mathrm{A}}, \qquad \Theta_{\mathrm{A}} \cap \Theta_{\mathrm{M}} = \varnothing.$$

- AdamW owns embeddings, head, mHC biases/gates, RMSNorm weights.
- Every other matrix is Muon-owned: momentum, Nesterov look-ahead, hybrid Newton–Schulz, shape-aware rescale, decay, subtract.
- Muon reshapes the *proposed update*; it never projects the stored weight W onto an orthogonality constraint.

[REPORTED: DeepSeek-V4 technical report]

Eight aggressive iterations, then two for cleanup

$$Z_r = aZ_{r-1} + b(Z_{r-1}Z_{r-1}^\top)Z_{r-1} + c(Z_{r-1}Z_{r-1}^\top)^2 Z_{r-1}, \qquad O_u = \gamma_{\mathrm{M}} \sqrt{\max(n, m)}\, O'_u.$$

Phase	Coefficients (a, b, c)	Job
Aggressive $\times 8$	$(3.4445, -4.7750, 2.0315)$	move channels fast; may overshoot

Eight aggressive iterations, then two for cleanup

$$Z_r = aZ_{r-1} + b(Z_{r-1}Z_{r-1}^\top)Z_{r-1} + c(Z_{r-1}Z_{r-1}^\top)^2 Z_{r-1}, \qquad O_u = \gamma_{\mathrm{M}}\sqrt{\max(n,m)}\, O'_u.$$

Phase	Coefficients (a, b, c)	Job
Aggressive $\times 8$	$(3.4445, -4.7750, 2.0315)$	move channels fast; may overshoot
Cleanup $\times 2$	$(2, -1.5, 0.5)$	pull channels near the polar fixed point 1

Eight aggressive iterations, then two for cleanup

$$Z_r = aZ_{r-1} + b(Z_{r-1}Z_{r-1}^\top)Z_{r-1} + c(Z_{r-1}Z_{r-1}^\top)^2 Z_{r-1}, \qquad O_u = \gamma_{\mathrm{M}}\sqrt{\max(n,m)}\, O'_u.$$

Phase	Coefficients (a, b, c)	Job
Aggressive $\times 8$	$(3.4445, -4.7750, 2.0315)$	move channels fast; may overshoot
Cleanup $\times 2$	$(2, -1.5, 0.5)$	pull channels near the polar fixed point 1

On the chapter's 4×3 rectangular example, the three singular channels reach $(1.1343, 0.6818, 0.7078)$ after 8 steps and $(1.00019, 1.00054, 1.00031)$ after both cleanup steps — orthogonality residual 1.08×10^{-3}. [DERIVED: book's rectangular Muon worked example, Section 4]

From concept to code: the channel polynomial, executable

The book's dependency-light checkpoint makes the displayed scalar polynomial $\phi(s) = as + bs^3 + cs^5$ runnable, exactly as the chapter states it — not a DeepSeek production optimizer.

from `code/chapter_deepseek_v4_training/muon_channel_checkpoint.py`

```
AGGRESSIVE = (3.4445, -4.7750, 2.0315)
CLEANUP = (2.0, -1.5, 0.5)

def step(values, coefficients):
    a, b, c = coefficients
    return tuple(a * s + b * s**3 + c * s**5 for s in values)
```

From concept to code: the channel polynomial, executable

The book's dependency-light checkpoint makes the displayed scalar polynomial $\phi(s) = as + bs^3 + cs^5$ runnable, exactly as the chapter states it — not a DeepSeek production optimizer.

from `code/chapter_deepseek_v4_training/muon_channel_checkpoint.py`

```
AGGRESSIVE = (3.4445, -4.7750, 2.0315)
CLEANUP = (2.0, -1.5, 0.5)

def step(values, coefficients):
    a, b, c = coefficients
    return tuple(a * s + b * s**3 + c * s**5 for s in values)
```

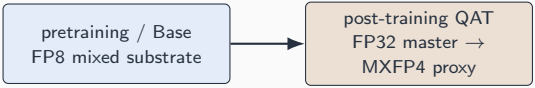
Eight calls with `AGGRESSIVE`, then two with `CLEANUP`, reproduce the rectangular trace's $r = 8$ and $r = 10$ channel values and the scaled-RMS assertion 0.180062. [DERIVED: book companion checkpoint, `muon_channel_checkpoint.py`]

Low-precision QAT

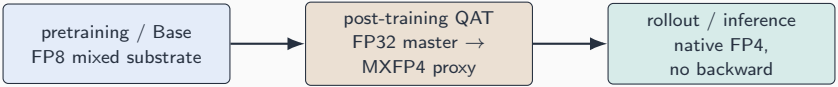
One weight, staged arithmetic across three phases

pretraining / Base
FP8 mixed substrate

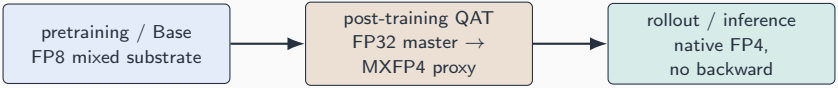
One weight, staged arithmetic across three phases



One weight, staged arithmetic across three phases



One weight, staged arithmetic across three phases



QAT covers exactly two paths: MoE expert weights, and the CSA lightning indexer's query–key path.

$$W_{32} \xrightarrow{Q_{\mathrm{MXFP4}}} (q_4, s_4) \xrightarrow{D_{\mathrm{E4M3}}} \widetilde{W}_8 \xrightarrow{\text{forward}} \mathcal{L}, \qquad \nabla_{W_{32}}\mathcal{L} \xleftarrow{\mathrm{STE}} \nabla_{\widetilde{W}_8}\mathcal{L}.$$

[REPORTED: DeepSeek-V4 technical report]

Two toy traces, two different oracles

- **Expert weight:** $W_{32} = (0.2, 0.7, 1.1, 2.8)$,
  $s_4 = 0.5 \to q_4 = (0.5, 1.5, 2, 6) \to \widetilde{W}_8 = (0.25, 0.75, 1, 3)$. All-ones dot product moves $4.8 \to 5.0$: a 0.2 output deviation, with the STE gradient still landing on W_{32}.

Two toy traces, two different oracles

- **Expert weight:** $W_{32} = (0.2, 0.7, 1.1, 2.8)$,
  $s_4 = 0.5 \to q_4 = (0.5, 1.5, 2, 6) \to \widetilde{W}_8 = (0.25, 0.75, 1, 3)$. All-ones dot product moves $4.8 \to 5.0$: a 0.2 output deviation, with the STE gradient still landing on W_{32}.
- **CSA indexer:** reference scores $1.41, 1.36$ become $1.25, 1.00$ under MXFP4; top-1 recall still selects k_1 even though the score error is $0.16/0.36$.

Two toy traces, two different oracles

- **Expert weight:** $W_{32} = (0.2, 0.7, 1.1, 2.8)$, $s_4 = 0.5 \to q_4 = (0.5, 1.5, 2, 6) \to \widetilde{W}_8 = (0.25, 0.75, 1, 3)$. All-ones dot product moves $4.8 \to 5.0$: a 0.2 output deviation, with the STE gradient still landing on W_{32}.
- **CSA indexer:** reference scores $1.41, 1.36$ become $1.25, 1.00$ under MXFP4; top-1 recall still selects k_1 even though the score error is $0.16/0.36$.

Numerical closeness and selected-entry identity are related but serve different oracles — a test needs both. The report attributes a $2\times$ selector speedup and 99.7% KV-entry recall to this path. [REPORTED: DeepSeek-V4 technical report] [DERIVED: book's toy expert and indexer traces, Section 5]

Post-training: specialists, GRM, and on-policy distillation

One actor, two roles: policy and judge

$$y_i \sim \pi_\theta(\cdot \mid x), \quad j_i \sim J_\theta(\cdot \mid x, y_i, r), \quad \mathcal{R}_i = \mathcal{C}(j_i; x, y_i, r), \quad A_i = \mathcal{G}(\mathcal{R}_i; \mathcal{R}_{1:G}).$$

- Rule checks score easy-to-verify tasks; a Generative Reward Model (GRM) scores hard-to-verify ones via rubric-guided evaluation.
- The same actor network plays the policy role π_θ and the generative judging role J_θ; RL jointly optimizes both.
- The bridge $\mathcal{C}$ and the group rule $\mathcal{G}$ are not published — the diagram names the public dataflow, not a recovered private recipe.

[REPORTED: DeepSeek-V4 technical report]

On-policy, full-vocabulary reverse KL — one learner, not an average

$$\mathcal{L}_{\mathrm{OPD}}(\theta; x, y) = \sum_{i=1}^{N} \omega_i \sum_{t} \sum_{v \in \mathcal{V}} \pi_\theta(v \mid c_t) \log \frac{\pi_\theta(v \mid c_t)}{\pi_{E_i}(v \mid c_t)}.$$

- Contexts c_t come from the current learner (*on-policy*); every vocabulary logit contributes (*full-vocabulary*).
- Toy check: learner $q = (0.6, 0.4)$, fixed teacher $p = (0.9, 0.1)$ gives $D_{\mathrm{KL}}(q \| p) \approx 0.311$ at one position.
- More than ten fixed teachers supply targets; only the learner's parameters move.

[REPORTED: DeepSeek-V4 technical report] [DERIVED: book's one-position KL trace, Section 6]

Evaluation surfaces and open questions

Quality rows are not a deployment frontier

Definition (Latency-resource-quality frontier)
A configuration is dominated if another candidate uses no more resource, is no slower, and is no less accurate, with at least one strict improvement. Nondominated candidates form the frontier.

Toy tuples (resource, p95 latency, accuracy): $A = (0.80, 1.2, 0.94)$, $B = (1.00, 1.1, 0.95)$, $C = (0.70, 1.5, 0.93)$, $D = (1.20, 1.4, 0.95)$. D is dominated by B; the frontier is $\{A, B, C\}$.

[DERIVED: book's dominance miniature, Section 7]

Pro is the larger operating envelope — deeper stack, more routed experts, larger CSA selector budget. [INSPECTED: Pro/Flash configuration]

Six matched benchmark pairs, and the interaction map

Benchmark	Flash	Pro	Pro − Flash
MMLU-Pro	86.2	87.5	1.3
Terminal Bench 2.0	56.9	67.9	11.0
MRCR 1M	78.7	83.5	4.8

Pro leads every displayed row; the gap is largest on Terminal Bench 2.0. These are quality coordinates only — no latency or resource measurement accompanies them, so they cannot establish frontier dominance. [REPORTED: DeepSeek-V4 technical report; Pro/Flash model cards]

Summary and what carries forward

What this chapter established

- One outer recipe (32T/33T tokens) carries a dense-to-sparse attention curriculum through indexer warm-up to sparse continuation.
- Compressed-attention context parallelism repairs one boundary, then all-gathers summaries — never a full KV circuit.
- Muon and AdamW partition every parameter exactly once; Muon reshapes the update, never the stored weight.
- QAT is staged and path-scoped: FP32 master, MXFP4/E4M3 forward proxy, native FP4 only after training ends.
- Post-training builds fixed specialists, judges hard tasks with an actor-owned GRM, then consolidates with on-policy full-vocabulary reverse KL.

[REPORTED: training stages and policy evaluation] [INSPECTED: Pro/Flash configuration] [DERIVED: routing, Muon, QAT, GRM, and OPD traces]

Matched optimizer, precision, curriculum, reward, and consolidation ablations remain unavailable — not inferred.

Concept-to-code map for the chapter

- **pretraining schedule, curriculum, ownership** → `DeepSeek-V4 technical report (DeepSeek_V4.pdf)`
- **rectangular Muon channel trace** → `code/chapter_deepseek_v4_training/muon_channel_checkpoint.py`
- **Muon design and hybrid Newton–Schulz lineage** → `arXiv:2502.16982, kellerjordan.github.io/posts/muon`
- **operating-envelope comparison** → `Pro/Flash model cards`

No DeepSeek-V4 training repository is open-sourced at the time of writing — this chapter's evidence lives in the report, the pinned model cards, and the book's own worked traces.

Name the state, name the owner, name what crosses to the next stage.

Next: the agent-interfaces chapter follows the policy into tool protocols and the DeepSeek Harness; the runtime chapter follows it into cache state and speculative decoding.