DeepSeek-V4 Architecture

Deep Dive into DeepSeek — Chapter 12

Yin Yang

Foundation Books Project

The one rule of this chapter

V4 changes *what attention compresses* and *how the residual stream is mixed*.
It does not retire DeepSeekMoE, MTP, or the causal contract.

- Two operators carry the headline story: hybrid CSA/HCA attention over compressed KV state, and mHC's doubly-stochastic residual mixing.
- Everything else in the forward pass — DeepSeekMoE routing, multi-token prediction — is inherited from V3, unchanged by report.
- Training, agent behavior, and serving are not this chapter: they get their own chapters so optimizer state, policy behavior, and cache layout are never mistaken for checkpoint sublayers.

The one rule of this chapter

V4 changes *what attention compresses* and *how the residual stream is mixed*.
It does not retire DeepSeekMoE, MTP, or the causal contract.

- Two operators carry the headline story: hybrid CSA/HCA attention over compressed KV state, and mHC's doubly-stochastic residual mixing.
- Everything else in the forward pass — DeepSeekMoE routing, multi-token prediction — is inherited from V3, unchanged by report.
- Training, agent behavior, and serving are not this chapter: they get their own chapters so optimizer state, policy behavior, and cache layout are never mistaken for checkpoint sublayers.

Goal: follow one token through a V4 block — hybrid attention, mHC residual mixing, and the boundaries where operators must agree — and keep every number's evidence status visible.

Roadmap

1. V4-Pro and V4-Flash in the family

Roadmap

1. V4-Pro and V4-Flash in the family
2. Hybrid attention: CSA, HCA, and DSA

Roadmap

1. V4-Pro and V4-Flash in the family
2. Hybrid attention: CSA, HCA, and DSA
3. From residual connections to mHC

Roadmap

1. V4-Pro and V4-Flash in the family
2. Hybrid attention: CSA, HCA, and DSA
3. From residual connections to mHC
4. Architecture interactions and open questions

Roadmap

1. V4-Pro and V4-Flash in the family
2. Hybrid attention: CSA, HCA, and DSA
3. From residual connections to mHC
4. Architecture interactions and open questions
5. Summary and what carries forward

V4-Pro and V4-Flash in the family

Three coupled pressures, one preview list

- Long context must not inherit dense attention cost.
- Sparse capacity must not become dense per-token work.
- Deeper learned residual pathways must remain stable through depth.
- Specialized post-training must still yield one deployable policy.

The official report answers with three headline upgrades: hybrid CSA/HCA attention, mHC residual mixing, and the Muon optimizer [REPORTED: DeepSeek-V4 technical report].

- That is an upgrade hierarchy, not the entire model.
- V4 retains the Transformer, DeepSeekMoE, and MTP lineage — the report says the MTP configuration is unchanged from V3.

Pro and Flash: reported scale

Quantity	V4-Pro	V4-Flash
Total parameters	1.6T	284B

[REPORTED: DeepSeek API preview release note]

- The preview note names token-wise compression plus DSA as a central innovation for the whole family.
- These numbers establish scale and capacity. They do not determine workload quality or serving latency.

Pro and Flash: reported scale

Quantity	V4-Pro	V4-Flash
Total parameters	1.6T	284B
Active parameters per token	49B	13B

[REPORTED: DeepSeek API preview release note]

- The preview note names token-wise compression plus DSA as a central innovation for the whole family.
- These numbers establish scale and capacity. They do not determine workload quality or serving latency.

Pro and Flash: reported scale

Quantity	V4-Pro	V4-Flash
Total parameters	1.6T	284B
Active parameters per token	49B	13B
Context window	1,048,576 positions	

[REPORTED: DeepSeek API preview release note]

- The preview note names token-wise compression plus DSA as a central innovation for the whole family.
- These numbers establish scale and capacity. They do not determine workload quality or serving latency.

Structural budgets, side by side

Field	V4-Flash	V4-Pro
Decoder stack	43 layers, hidden 4096	61 layers, hidden 7168

[INSPECTED: DeepSeek-V4-Pro / DeepSeek-V4-Flash `config.json`]

Structural budgets, side by side

Field	V4-Flash	V4-Pro
Decoder stack	43 layers, hidden 4096	61 layers, hidden 7168
CSA selector top-k	`index_topk=512`	`index_topk=1024`

[INSPECTED: DeepSeek-V4-Pro / DeepSeek-V4-Flash `config.json`]

Structural budgets, side by side

Field	V4-Flash	V4-Pro
Decoder stack	43 layers, hidden 4096	61 layers, hidden 7168
CSA selector top-k	`index_topk=512`	`index_topk=1024`
MoE experts	256 routed, 6 active	384 routed, 6 active

[INSPECTED: DeepSeek-V4-Pro / DeepSeek-V4-Flash `config.json`]

Structural budgets, side by side

Field	V4-Flash	V4-Pro
Decoder stack	43 layers, hidden 4096	61 layers, hidden 7168
CSA selector top-k	`index_topk=512`	`index_topk=1024`
MoE experts	256 routed, 6 active	384 routed, 6 active
mHC fields	`hc_mult=4`, `hc_sinkhorn_iters=20`	

[INSPECTED: DeepSeek-V4-Pro / DeepSeek-V4-Flash `config.json`]

Structural budgets, side by side

Field	V4-Flash	V4-Pro
Decoder stack	43 layers, hidden 4096	61 layers, hidden 7168
CSA selector top-k	`index_topk=512`	`index_topk=1024`
MoE experts	256 routed, 6 active	384 routed, 6 active
mHC fields	`hc_mult=4`, `hc_sinkhorn_iters=20`	

[INSPECTED: DeepSeek-V4-Pro / DeepSeek-V4-Flash `config.json`]

Both variants share the same mechanism family. They differ in depth, width, expert pool, and selector budget — not in kind.

Active parameter ratio

Definition
For total parameters P_{total} and per-token active parameters P_{active},

$$\rho = \frac{P_{\text{active}}}{P_{\text{total}}}.$$

Applying it to the reported counts: $\rho_{\text{Pro}} = 49/1600 \approx 3.1\%$ and $\rho_{\text{Flash}} = 13/284 \approx 4.6\%$

[DERIVED: reported total/active counts].

Pro's smaller ratio does not make it cheaper: it still activates 49B parameters per token, on a wider stack with a larger selector. Ratios describe conditional activation, not a product ranking.

Hybrid attention: CSA, HCA, and DSA

Three resolutions across the stack

SWA
exact recent window

preserves nearby dependencies; runs alongside every layer

Three resolutions across the stack

SWA
exact recent window

preserves nearby dependencies; runs alongside every layer

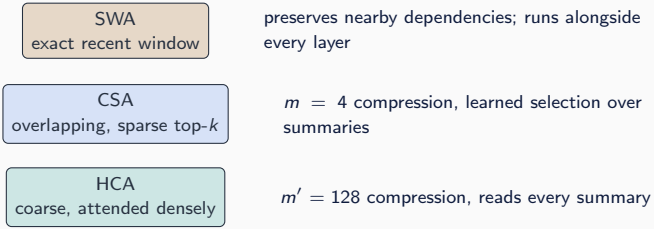
CSA
overlapping, sparse top-k

$m = 4$ compression, learned selection over summaries

Three resolutions across the stack

SWA exact recent window	preserves nearby dependencies; runs alongside every layer
CSA overlapping, sparse top-k	$m = 4$ compression, learned selection over summaries
HCA coarse, attended densely	$m' = 128$ compression, reads every summary

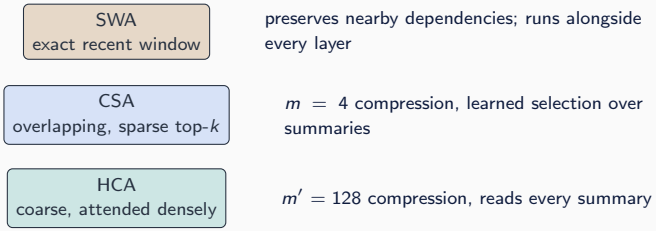
Three resolutions across the stack



A hybrid layer schedules CSA *or* HCA as its core path; SWA runs in parallel with either choice

[REPORTED: DeepSeek-V4 technical report].

Three resolutions across the stack



A hybrid layer schedules CSA *or* HCA as its core path; SWA runs in parallel with either choice [REPORTED: DeepSeek-V4 technical report].

Compression changes representation. Selection changes which represented entries participate. Those are different knobs.

CSA and HCA, defined against each other

	CSA	HCA
Pooling	overlapping $2m$-token span, stride m	non-overlapping m'-token groups
Rate	$m = 4$	$m' = 128$
After pooling	DSA top-k selection	dense attention over all entries

[REPORTED: DeepSeek-V4 technical report]

CSA trades a small, overlapping compression rate for learned sparsity. HCA trades a much larger compression rate for exhaustive, dense read. Both still cache shared-KV entries and run alongside SWA.

DSA, specialized to V4

- Recall DeepSeek Sparse Attention from V3.2: a lightning indexer ranks prior *positions*; top-k selection admits their latent KV entries.

DSA, specialized to V4

- Recall DeepSeek Sparse Attention from V3.2: a lightning indexer ranks prior *positions*; top-k selection admits their latent KV entries.
- V4 applies the same selection rule *inside CSA*, but after KV compression.

Its indexer now ranks compressed entries that summarize source spans, not the per-position latent entries selected in V3.2 [REPORTED: DeepSeek-V4 technical report; V3.2-Exp release note].

DSA, specialized to V4

- Recall DeepSeek Sparse Attention from V3.2: a lightning indexer ranks prior *positions*; top-k selection admits their latent KV entries.
- V4 applies the same selection rule *inside CSA*, but after KV compression.

Its indexer now ranks compressed entries that summarize source spans, not the per-position latent entries selected in V3.2 [REPORTED: DeepSeek-V4 technical report; V3.2-Exp release note].

That shift means selector recall and compression fidelity become two separate things to test — a selector can pick a legal index while the underlying compressed entry is a poor summary, or vice versa.

From residual connections to mHC

The rigid single lane, and the risk of relaxing it

Ordinary residual: $x_{l+1} = x_l + \mathcal{F}_l(x_l)$. The identity route is never multiplied by a learned map.

Hyper-Connections (HC) expand the stream to n lanes and learn pre-, post-, and residual maps around each sublayer — but an *unconstrained* residual map $\mathcal{H}_l^{\text{res}}$ composes across depth:

$$\mathcal{M}_{l \to L} = \mathcal{H}_{L-1}^{\text{res}} \cdots \mathcal{H}_l^{\text{res}}.$$

As a scalar illustration, a gain of only 1.05 repeated across 64 layers becomes $1.05^{64} \approx 22.7$ [DERIVED: scalar illustration, not a paper measurement].

The danger is a product, not one alarming layer. mHC is the fix.

mHC: the doubly stochastic constraint

Definition (doubly stochastic residual mixer)
A nonnegative matrix P with $\sum_j P_{i,j} = 1$ and $\sum_i P_{i,j} = 1$ for every row i and column j.

- mHC keeps HC's expanded residual lanes but constrains $\mathcal{H}_l^{\text{res}}$ to this manifold [REPORTED: DeepSeek-V4 technical report; mHC paper].

mHC: the doubly stochastic constraint

Definition (doubly stochastic residual mixer)
A nonnegative matrix P with $\sum_j P_{i,j} = 1$ and $\sum_i P_{i,j} = 1$ for every row i and column j.

- mHC keeps HC's expanded residual lanes but constrains $\mathcal{H}_l^{\text{res}}$ to this manifold [REPORTED: DeepSeek-V4 technical report; mHC paper].
- The product of doubly stochastic matrices stays doubly stochastic — the closure property that makes a *deep stack* safe.

mHC: the doubly stochastic constraint

Definition (doubly stochastic residual mixer)
A nonnegative matrix P with $\sum_j P_{i,j} = 1$ and $\sum_i P_{i,j} = 1$ for every row i and column j.

- mHC keeps HC's expanded residual lanes but constrains $\mathcal{H}_l^{\text{res}}$ to this manifold [REPORTED: DeepSeek-V4 technical report; mHC paper].
- The product of doubly stochastic matrices stays doubly stochastic — the closure property that makes a *deep stack* safe.
- Birkhoff intuition: the mixer is a convex combination of lane permutations, so it can reroute without an arbitrary gain path.

mHC: the doubly stochastic constraint

Definition (doubly stochastic residual mixer)
A nonnegative matrix P with $\sum_j P_{i,j} = 1$ and $\sum_i P_{i,j} = 1$ for every row i and column j.

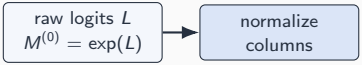
- mHC keeps HC's expanded residual lanes but constrains $\mathcal{H}_l^{\text{res}}$ to this manifold [REPORTED: DeepSeek-V4 technical report; mHC paper].
- The product of doubly stochastic matrices stays doubly stochastic — the closure property that makes a *deep stack* safe.
- Birkhoff intuition: the mixer is a convex combination of lane permutations, so it can reroute without an arbitrary gain path.

This is residual-stream mixing. It is not a new attention mechanism and not a retrieval improvement.

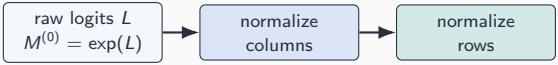
Reaching the constraint: Sinkhorn balancing

raw logits L
$M^{(0)} = \exp(L)$

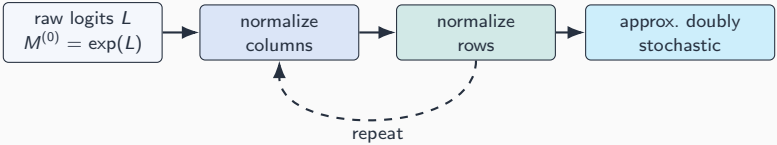
Reaching the constraint: Sinkhorn balancing



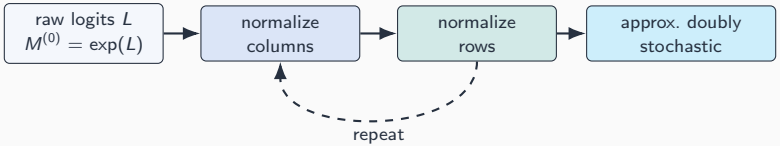
Reaching the constraint: Sinkhorn balancing



Reaching the constraint: Sinkhorn balancing

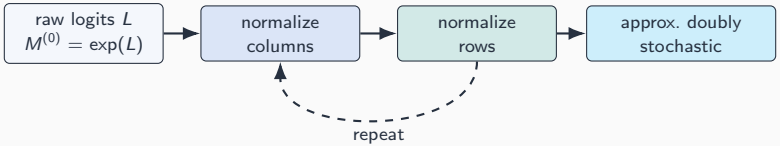


Reaching the constraint: Sinkhorn balancing



Column normalization fixes column sums but disturbs row sums; row normalization then fixes rows and disturbs columns. Alternating both converges toward doubly stochastic [REPORTED: mHC paper].

Reaching the constraint: Sinkhorn balancing



Column normalization fixes column sums but disturbs row sums; row normalization then fixes rows and disturbs columns. Alternating both converges toward doubly stochastic [REPORTED: mHC paper].

Real kernels stop after `hc_sinkhorn_iters=20` alternations [INSPECTED: Pro/Flash `config.json`] — an approximate projection, not a symbolic guarantee that every sum is exactly one.

Composed across two sublayers: a numeric certificate

For an attention sublayer then a feed-forward sublayer, the direct residual route composes as

$$M = P_1 P_0.$$

- In a constructed four-lane, two-sublayer trace: $M\mathbf{1} = \mathbf{1}$, $\mathbf{1}^\top M = \mathbf{1}^\top$, and $\|M\|_1 = \|M\|_\infty = 1$ [DERIVED: worked architectural trace, not recovered V4 weights].
- A didactic unconstrained control $Q_l = 1.05\, P_l$ already reaches $\|Q_1 Q_0\|_1 = 1.1025$ after only two sublayers.

At V4 scale, the official paper's 120-layer diagnostic reports unconstrained HC's composite gain approaching 3000, while mHC stays around 1.6, at a reported 6.7% marginal training overhead [REPORTED: mHC paper].

Those are training diagnostics, not a serving-latency or retrieval guarantee.

Architecture interactions and open questions

Where operators must agree

Boundary	Shared state that must agree
CSA/HCA + DSA	compressed entries, selector scores/indices, exact SWA state, positions
mHC + depth	four residual lanes, constrained mixing matrices
Attention + mHC	selected attention output, lane reads and writes
mHC + MoE	feed-forward input lanes, routed-token identity, residual write

A shape-correct implementation can still be wrong: legal tensors, wrong compressed span, selection, or lane boundary. [OPEN: joint architecture-interaction ablations]

Engram stays adjacent

- The released V4-Pro/Flash mechanism set is backed by the report and artifacts: mHC, CSA, HCA, DSA, DeepSeekMoE, MTP.
- Those sources mention Engram only as a future direction, not a released component.

[REPORTED: DeepSeek-V4 technical report; Pro/Flash model cards; Engram repository/paper]

A source-family resemblance does not promote adjacent research into released architecture.

Summary and what carries forward

What this chapter established

- Hybrid CSA/HCA attention over compressed KV state, with an exact SWA path and a DSA selector specialized to compressed entries.

[REPORTED: architecture and evaluation] [INSPECTED: Pro/Flash artifact and configuration surfaces] [DERIVED: shape, compression, and residual traces]

What this chapter established

- Hybrid CSA/HCA attention over compressed KV state, with an exact SWA path and a DSA selector specialized to compressed entries.
- mHC replaces one residual stream with four constrained lanes, across both attention and feed-forward sublayers.

[REPORTED: architecture and evaluation] [INSPECTED: Pro/Flash artifact and configuration surfaces] [DERIVED: shape, compression, and residual traces]

What this chapter established

- Hybrid CSA/HCA attention over compressed KV state, with an exact SWA path and a DSA selector specialized to compressed entries.
- mHC replaces one residual stream with four constrained lanes, across both attention and feed-forward sublayers.
- Pro and Flash instantiate the same mechanism family at different width, depth, expert, and selector budgets.

[REPORTED: architecture and evaluation] [INSPECTED: Pro/Flash artifact and configuration surfaces] [DERIVED: shape, compression, and residual traces]

What this chapter established

- Hybrid CSA/HCA attention over compressed KV state, with an exact SWA path and a DSA selector specialized to compressed entries.
- mHC replaces one residual stream with four constrained lanes, across both attention and feed-forward sublayers.
- Pro and Flash instantiate the same mechanism family at different width, depth, expert, and selector budgets.
- DeepSeekMoE and MTP are inherited, not new V4 architecture.

[REPORTED: architecture and evaluation] [INSPECTED: Pro/Flash artifact and configuration surfaces] [DERIVED: shape, compression, and residual traces]

What this chapter established

- Hybrid CSA/HCA attention over compressed KV state, with an exact SWA path and a DSA selector specialized to compressed entries.
- mHC replaces one residual stream with four constrained lanes, across both attention and feed-forward sublayers.
- Pro and Flash instantiate the same mechanism family at different width, depth, expert, and selector budgets.
- DeepSeekMoE and MTP are inherited, not new V4 architecture.

[REPORTED: architecture and evaluation] [INSPECTED: Pro/Flash artifact and configuration surfaces] [DERIVED: shape, compression, and residual traces]

None of these structural facts alone ranks quality or serving speed — matched ablations remain open, not inferred.

Concept-to-code map for the chapter

- **Pro/Flash structural fields** → `DeepSeek-V4-Pro/config.json, DeepSeek-V4-Flash/config.json`
- **headline mechanism definitions** → `DeepSeek-V4 technical report (DeepSeek_V4.pdf)`
- **doubly stochastic residual mixer** → `mHC paper, arXiv:2512.24880`
- **released model class** → `DeepseekV4ForCausalLM, Pro/Flash model cards`

No DeepSeek-V4 inference repository is open-sourced at the time of writing — unlike V3, this chapter's evidence lives in the technical report and the pinned model-card configurations, not in inspectable kernel code.

Compress what attention reads. Constrain how the residual mixes.

Next: the training chapter builds these checkpoints — Muon, FP4 QAT, and specialist consolidation into one policy.