READ MODE · ANSWER FIRST

How DeepSeek-V4 architecture and training work

Explore DeepSeek-V4 mHC residuals, CSA and HCA attention, sparse experts, Muon pretraining, stability guards, and OPD.

CHEAT SHEET · 6 ESSENTIAL IDEAS

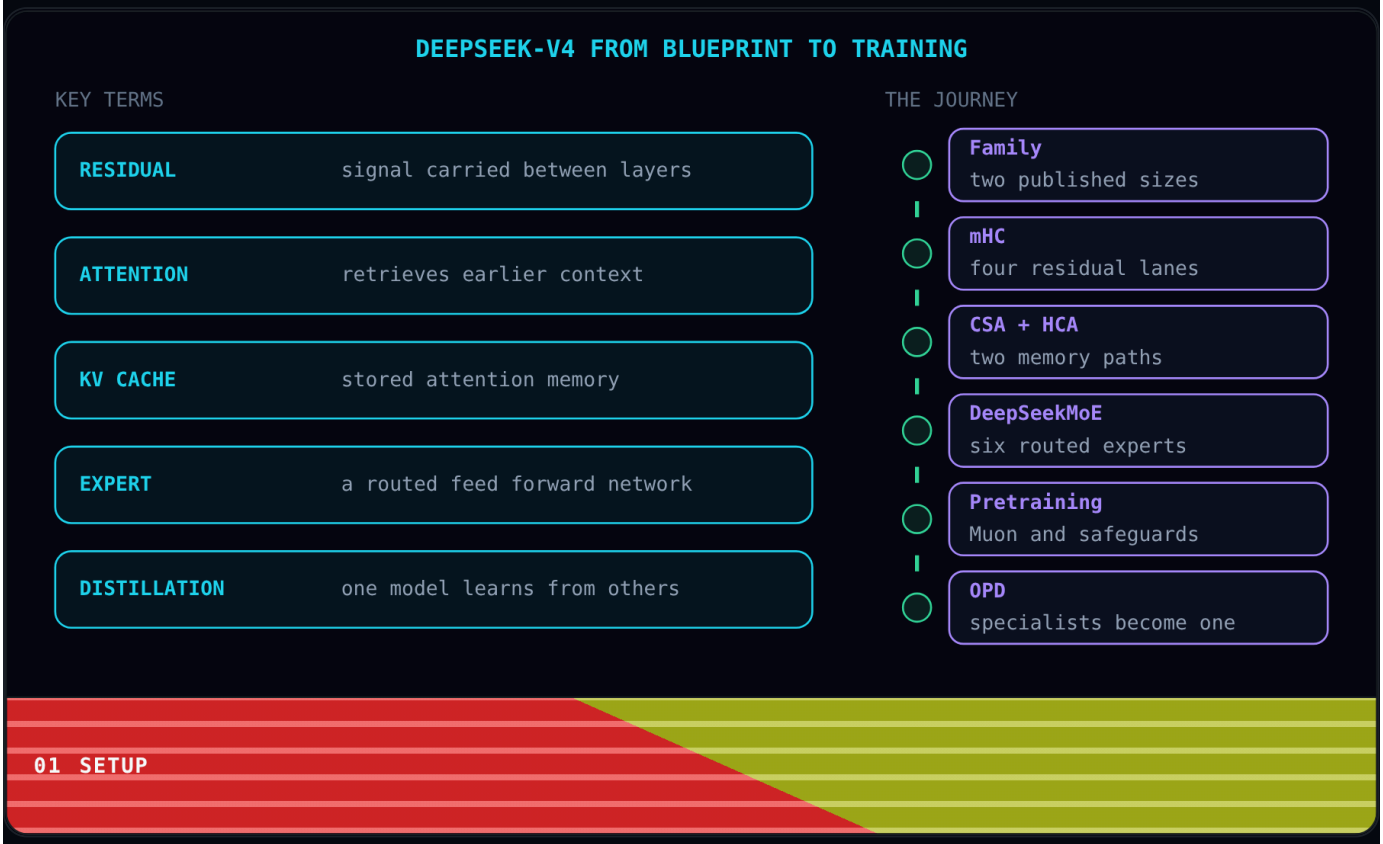
The whole story in 6 lines

DeepSeek-V4 co-designs residual flow, compressed attention, sparse experts, stable pretraining, and specialist distillation for...

1. Total scale and active scale are different in an MoE model. Only a small slice works on each token.
2. A doubly stochastic mixing matrix keeps every row and column normalized instead of letting signal scale drift.
3. CSA keeps selected detail after 4 to 1 compression, while HCA uses 128 to 1 compression and dense attention.
4. Routing turns a huge parameter pool into a sparse per-token path.
5. Long context arrives gradually, while anticipatory routing and SwiGLU clamping address instability when it appears.
6. The student learns from teacher output distributions on trajectories the student generated itself.

STAGE 1 OF 8

Setup



DeepSeek-V4 is a family of language models that can use up to 1,048,576 tokens at once. We will see how its design and training process work together.

A residual stream carries information between layers, while attention looks back at earlier tokens and an expert handles each token with a small feed forward network.

A KV cache stores information that attention may need again. Distillation teaches one student model from other models. These ideas connect design, pretraining, and post training.

We will study the model family, residuals, attention, experts, pretraining, and distillation, so let us begin with the two published model sizes.

STAGE 2 OF 8

One Blueprint, Two Sizes

02 ONE BLUEPRINT, TWO SIZES

We now know the key terms and DeepSeek-V4 comes in two sizes and Flash is smaller and Pro is larger, but both follow the same basic design.

Flash has 43 layers and a width of 4,096. Pro has 61 layers and a width of 7,168. More layers and wider layers make Pro much larger.

Switch the Model control to compare 256 routed experts in Flash with 384 in Pro, but notice that both still choose only 6 for each token.

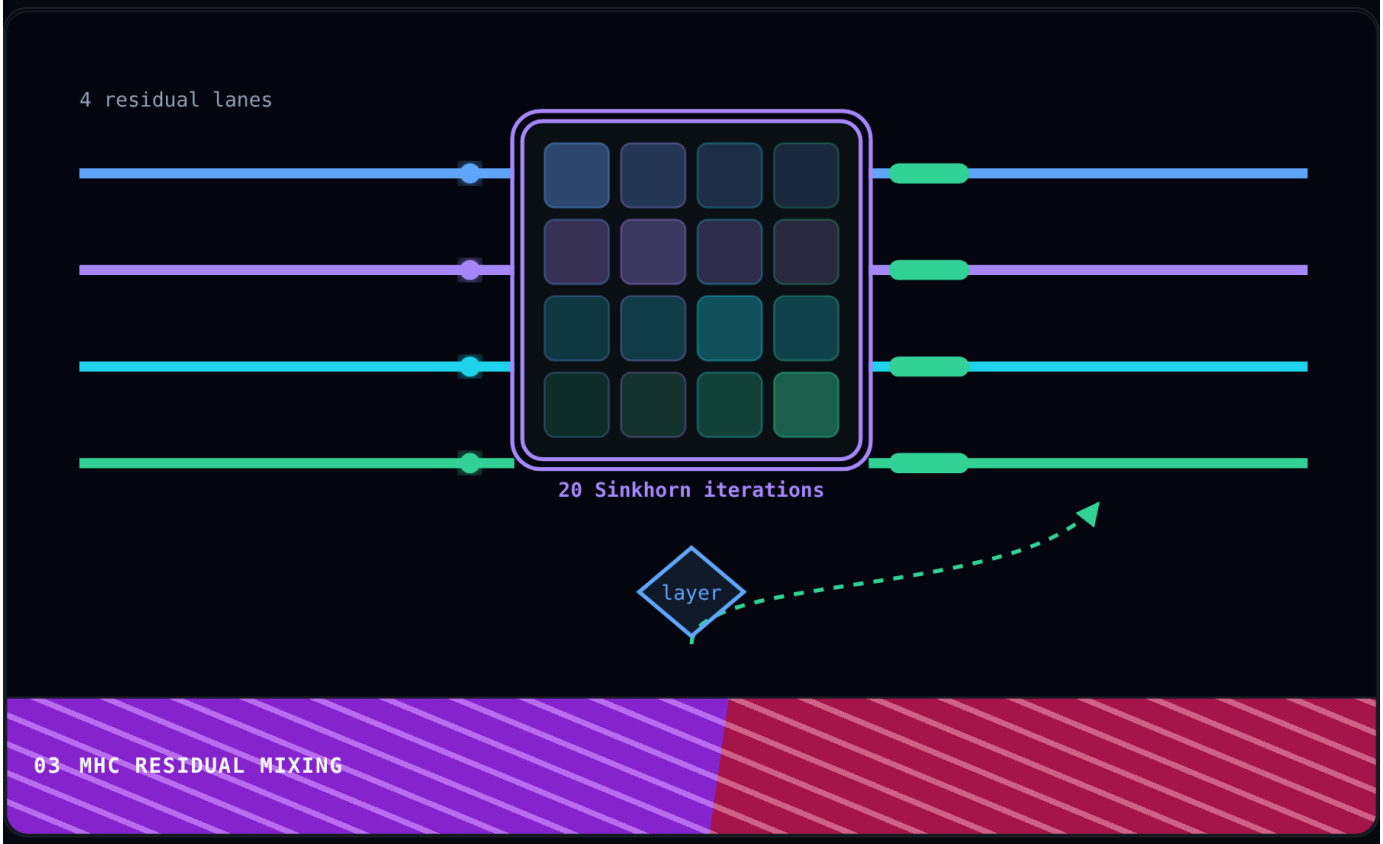
The total parameter bar is huge, so does one token really use all 284 billion or 1.6 trillion parameters?

No, because each token activates only 13 billion parameters in Flash or 49 billion in Pro, which is the bright slice inside the full model.

The model can add total capacity without making every token use all of it, so next we will follow the four residual lanes used by both models.

STAGE 3 OF 8

mHC Residual Mixing



Both models use four residual lanes and each layer reads a mixture of the four lanes and writes its result back into them.

A raw hyper connection may give the lanes uneven mixing weights and across many layers, some signals can then grow too large while others become weak.

Switch the Mapping control to compare Raw HC with mHC, then ask what changes when every row and column must keep a balanced total?

mHC runs 20 Sinkhorn iterations and they make every row and column add to the same total, so all four residual lanes keep balanced paths through the layer.

mHC keeps information flowing while reducing numerical drift, so next we will see how attention compresses a memory of one million tokens.

STAGE 4 OF 8

CSA Meets HCA

mHC kept signals stable between layers and the model now alternates CSA and HCA attention because each one solves a different long memory problem.

CSA compresses each group of 4 tokens into one entry. It selects 512 entries in Flash or 1,024 in Pro, plus the latest 128 local tokens.

HCA compresses each group of 128 tokens into one entry. It then uses dense attention on that shorter memory. Switch Model to compare the published layer schedules.

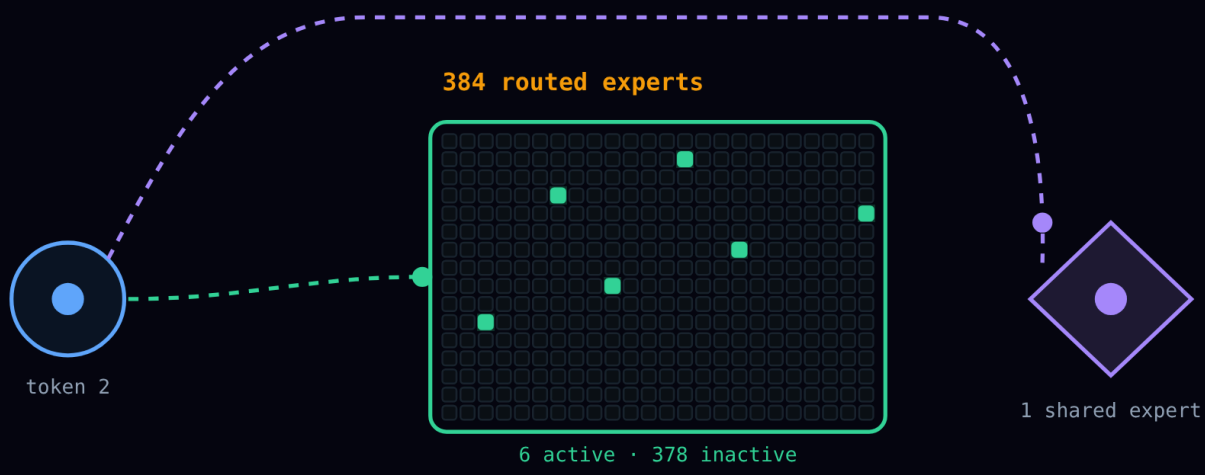
Set Context to 1M so both paths receive the same history, but how can one preserve detail while the other makes distant history cheaper?

CSA keeps many small selected groups and HCA turns every 128 token span into one dense capsule. Using both methods is the hybrid design.

Alternating the two paths balances detail and cost across the model, so next we will follow one token through the sparse expert network.

STAGE 5 OF 8

DeepSeekMoE Routing



05 DEEPSEEKMOE ROUTING

Hybrid attention produced a new state for one token and the feed forward part of the block now sends that state through DeepSeekMoE.

One shared expert always runs and the model can also choose from 256 routed experts in Flash or 384 in Pro.

Change Token to see another repeatable expert route and change Model to switch between the official pool sizes.

The pool has hundreds of experts, so does every token need to run through all of them?

No, because each token runs through the shared expert and exactly 6 routed experts, while the many dark experts remain available for other tokens.

Sparse routing keeps the work for one token far below the model's total capacity, so next we will see how Muon and safety guards train this large system.

STAGE 6 OF 8

Pretraining at Scale

06 PRETRAINING AT SCALE

Sparse experts save work, but training a trillion parameter MoE can produce sudden loss spikes. We will follow the training plan and its safety systems.

Muon updates most parameters with 0.95 momentum and 0.1 weight decay and AdamW still updates embeddings, prediction heads, and RMSNorm weights.

Training grows the context from 4K to 16K, 64K, and finally 1M tokens and sparse attention starts at 64K after dense attention warms up the model.

Switch Model to compare the training runs: Flash uses 32 trillion tokens and a 75.5 million maximum batch, while Pro uses 33 trillion and 94.4 million.

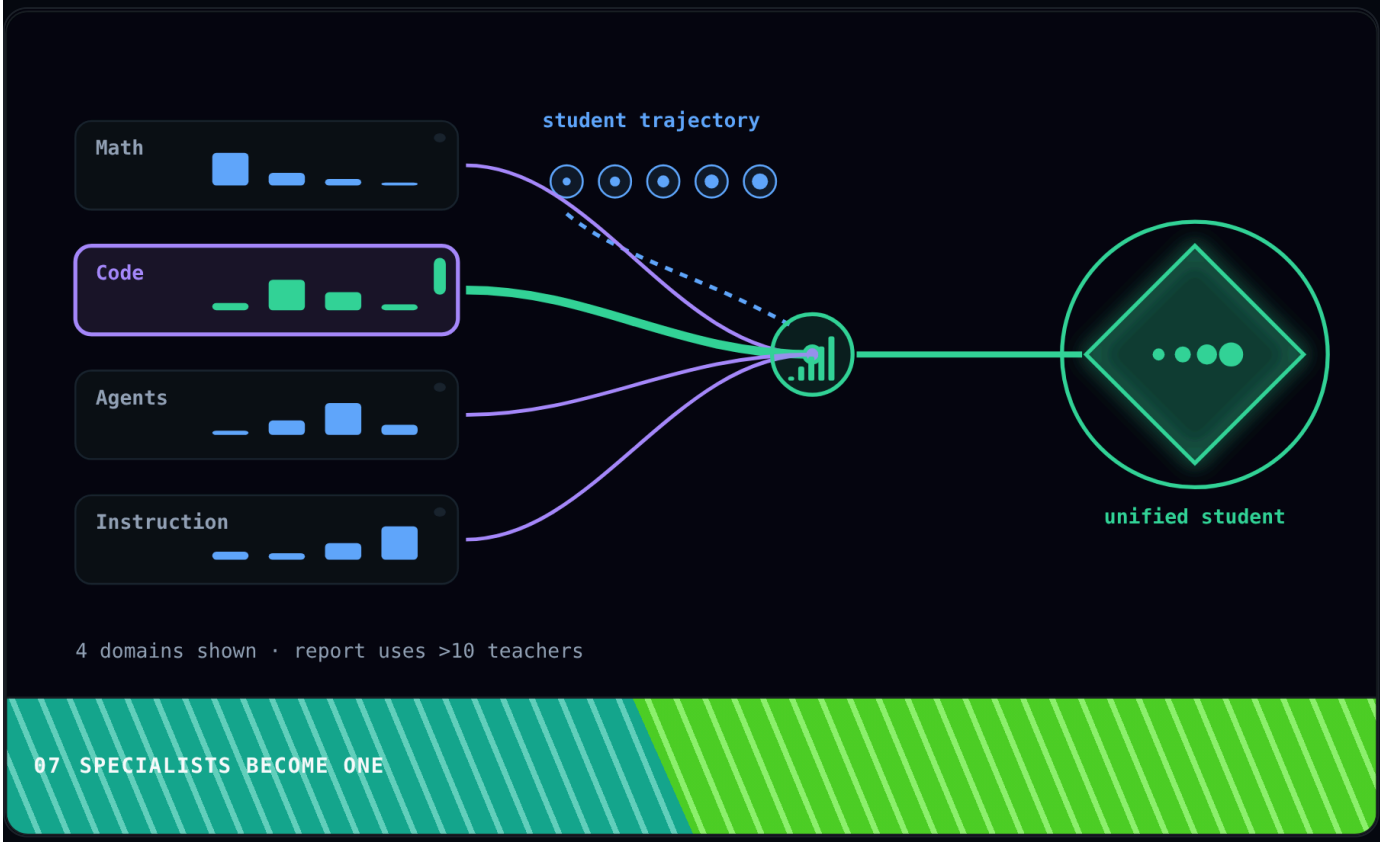
An unusual MoE value makes the loss jump, so if training only rolls back, can the same routing pattern cause another spike?

A rollback is not enough because anticipatory routing breaks the cycle and SwiGLU clamping limits extreme values to 10, so toggle Safeguards to compare both traces.

This staged pretraining uses 32 or 33 trillion tokens and reaches a 1M token context. Next we will combine separately trained specialists.

STAGE 7 OF 8

Specialists Become One



07 SPECIALISTS BECOME ONE

Pretraining built a strong base model and post training then creates separate specialists for math, code, agents, and instruction following.

Each specialist starts with fine tuning and it then learns from reinforcement learning that uses prompts and rewards from its own subject.

The student model generates its own answer path and switch Task to change which specialist's output distribution gets the most weight.

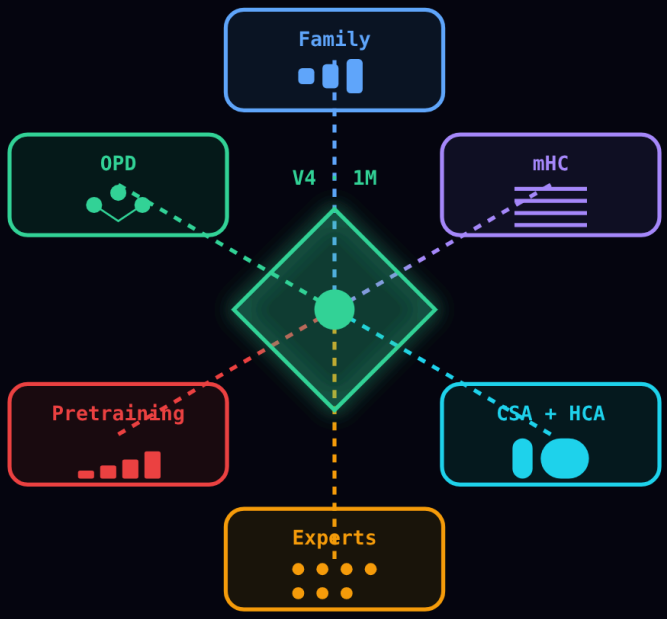
The full system has more than ten teacher models, so how can one student learn their skills without directly combining their model weights?

OPD gives the student each teacher's probability distribution across the full vocabulary and weighted teacher signals train one set of student parameters.

The specialists stay separate, but their behavior is distilled into one model and now let us step back and view the whole DeepSeek-V4 system.

STAGE 8 OF 8

Recap



08 RECAP

We started with Flash and Pro and they share one design, but Pro has more layers, wider layers, and more total parameters.

Then we studied mHC and its balanced mixing matrix moves information across four residual lanes without letting signal size drift.

Hybrid attention uses two kinds of memory. CSA keeps selected details and HCA makes distant history cheaper with 128 to 1 compression.

DeepSeekMoE adds a large pool of skills without using the whole pool and one shared expert and 6 routed experts handle each token.

Pretraining used Muon, safety guards, and a context schedule from 4K to 1M across 32 or 33 trillion tokens.

Post training built specialists before OPD taught their output distributions to one student and these values come from the official report and configs.

Stable residuals, compressed memory, sparse experts, staged pretraining, and distillation make this huge million token model practical to run.