Compute

Institutional Equity Research
April 24, 2026

DeepSeek-V4: Our Patience, Rewarded

For a year, we've been painstakingly waiting following DeepSeek-V3's infamous release (see our original takes), but now we are finally getting the official release of DeepSeek-V4, marking the latest flagship model from the Chinese open-source lab. Below, we detail some of our initial thoughts, with follow-ons to proceed after we have sufficient time to benchmark and assess both models ourselves.

The architecture is unsurprisingly ingenious, and we expect broad adoption of many of these techniques. DeepSeek-V4 replaces DSV3's Multi-Head Latent Attention (MLA) with a hybrid architecture that interleaves two novel techniques, Compressed Sparse Attention (CSA) which compresses the KV cache by 4x and selects only the most relevant entries via a learned indexer, and Heavily Compressed Attention (HCA), which aggressively compresses KV cache by 128x but lets every query see the entire context. The model alternates between a detailed-but-selective view and a complete-but-low resolution view of context across its layer stack, reducing KV cache to roughly 2% of a standard attention baseline at 1M tokens. Alongside this, DeepSeek uses Manifold-Constrained Hyper-Connections (mHC) to replace standard residual connections with an expanded residual stream whose mixing matrices are constrained to doubly stochastic matrices, guaranteeing stable signal propagation even across 61 layers. Further, we see the use of the Muon optimizer to replace AdamW for most parameters, using Newton-Schulz iterations to orthogonalize gradient updates for faster convergence, complemented by FP4 quantization-aware training and shared key-value attention that halves per-entry cache size. All together, this results in DeepSeek-V4-Pro using only 27% of DSV3.2's inference FLOPS and 10% of its KV cache.

Do not get it twisted though, compute and memory demand from the labs will not change. We believe that similar to the takeaways from DeepSeek-V3, many will fall for a head fake, given the seemingly drastic reductions in need for FLOPS and memory with DSV4. That being said, as we noted in our 2026 predictions piece, other labs will adopt or adapt these methods into their own architecture stack, however, the natural response from American labs and others will be to fill that headroom with longer contexts and reasoning chains, and more complex long-horizon agentic workloads.

DeepSeek-V4 is perhaps the strongest evidence yet that Chinese export controls are simultaneously working and insufficient. We'd argue this finding cuts directly against Jensen's argument on the recent Dwarkesh Podcast that China not only has enough compute, but that restricting chip sales merely cedes market share with little in return. Every notable architectural decision in the V4 technical paper is explicitly motivated by making 1M contexts feasible under compute constraints that American labs with abundant compute simply don't face. Even the technical report concedes that V4-Pro-Max trails frontier proprietary models on certain benchmarks because of the compute gap. And this is precisely the dynamic Dwarkesh identified on his podcast, which is that labs are bottlenecked by compute, and more compute translates to better models, thus Chinese labs want NVIDIA's chips. Jensen's position requires one to believe the compute lead doesn't meaningfully matter, but the V4 paper shows just how much architectural ingenuity is required to compensate for the compute differential and how that compensation remains insufficient. We'd argue that the deeper concern for export control hawks isn't what DeepSeek has achieved despite constraints, but rather that the paper reveals DeepSeek's readiness to capitalize on better hardware the moment it becomes available. The batch-invariant deterministic kernels, fused MoE mega-kernel, and TileLang compiler integration all suggest a stack built to immediately exploit any loosening of the compute bottleneck, which should give pause to anyone advocating for relaxed restrictions based on the premise that the horse has already left the stable.

See a technical breakdown of the architecture on the following pages.

INDUSTRY UPDATE

Price (4/25/26)
Industry: TECHNOLOGY

Alexander Platt
(503) 603-3045
AJPlatt@dadco.com

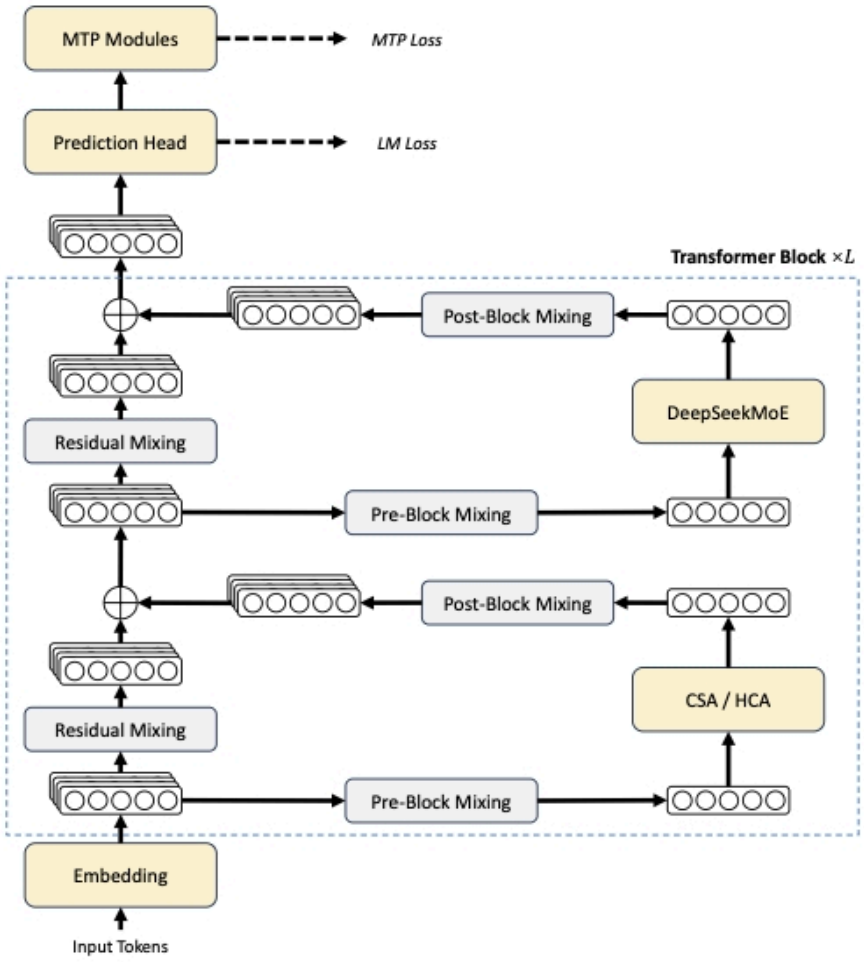
DaVinci Overview

D.A. Davidson's DaVinci initiative focuses our technical-oriented research, data-driven insights, and prescient think pieces under one unified framework. We note that for our DaVinci coverage of deep tech businesses, we employ an early-stage venture approach focusing on technical foundations, disruptive potential, and long-term strategic value, rather than near-term financial and valuation metrics given the unique growth trajectories of pre-inflection markets.

Architecture

DeepSeek-V4 retains the Transformer backbone and Multi-Token Prediction (MTP) modules from DeepSeek-V3, but introduces three fundamental upgrades (1) Manifold-Constrained Hyper-Connections (mHC), (2) a hybrid attention architecture combining CSA and HCA, and (3) the Muon optimizer. The DeepSeekMoE feed-forward architecture is carried forward with minor modifications, and MTP remains unchanged from the DSV3 architecture.

Figure 1: DeepSeek-V4 Architecture Overview



Source: "DeepSeek-V4: Towards Highly Efficient Million-Token Context Intelligence", DeepSeek-AI

DSV4 retains the DeepSeekMoE paradigm for feed-forward networks, which partitions experts into shared experts (universally active, providing consistent baseline contributions) and routed experts (dynamically activated per-token based on affinity scores). The activation function for computing expert affinity scores is changed from sigmoid to the square root of softplus. This provides a smoother, non-saturating gate that better differentiates between expert relevance levels as sigmoid saturates at extreme values, compressing the dynamic range of gating decisions, whereas sqrt(softplus) maintains a more even gradient across its range. Further, the auxiliary-loss-free load balancing strategy from DSV3 is retained, augmented by a lightweight sequence-wise balance loss. The core mechanism assigns each expert a bias term that is adjusted based on its workload in the preceding step. If an expert is overloaded, its bias decreases, making it less likely to be selected in subsequent iterations, while underloaded experts have their bias increased. This dynamic adjustment prevents routing collapse (a common MoE failure mode where only a small subset of experts are consistently utilized) without introducing the optimization conflicts that traditional auxiliary losses create.

Additionally, two other changes are noteworthy to call out. First, the constraint on the number of routing target nodes is removed, and the parallelism strategy is redesigned accordingly. Second, the initial dense FFN layers used in DSV3 are replaced with MoE layers that use Hash routing, where expert assignment is determined by a fixed hash function applied to the input token ID rather than a learned routing network. Hash routing eliminates routing overhead for the earliest layers, where learned representations may not yet be reliable enough for meaningful routing decisions.

The MTP strategy carries over from DSV3 without modification. As a quick refresher, at each position, the model predicts one additional future token beyond the standard next-token prediction. Each MTP module shares the embedding layer and output head with the main model, and retains the causal chain between predictions. Meaning the representation from predicting token t+1 informs the prediction of token t+2, rather than predicting both independently, and it's this kind of hierarchical structure that mirrors the natural progression of language. Additionally, MTP modules can be discarded during inference for simplicity or repurposed for speculative decoding to reduce generation latency.

Manifold-Constrained Hyper-Connections (mHC)

Standard residual connections in Transformers pass information between layers by simply adding the layer's output to its input. The output of layer l is x + f(x), where f(x) is whatever the layer computes. This additive structure is fundamental to training deep networks, but it constrains how information flows through the network as the residual stream has the same width as the hidden dimension, and information can only be added, not selectively mixed or transformed. Hyper-Connections (HC) generalize this by expanding the residual stream to carry multiple copies of the hidden dimension. Instead of one stream of width d, the residual state becomes a set of n_hc streams (n_hc = 4 in DSV4), each of width d. Three small learned mappings control how information flows (1) an input mapping that combines all n_hc streams into a single d-dimensional input for the actual layer, (2) a residual transformation that mixes information across the n_hc streams, and (3) an output mapping that distributes the layer's output back into the expanded stream. While HC has demonstrated potential in improving model performance, DeepSeek found that training frequently became numerically unstable when stacking many layers. The core problem here is the residual transformation matrix, which is the mapping that mixes information across the n_hc streams. When you multiply many of these matrices together (one per layer), the product can amplify signals uncontrollably, causing numerical explosion in both the forward pass and gradient computation.

The central innovation of mHC is constraining the residual transformation matrix to the set of doubly stochastic matrices. A doubly stochastic matrix is a non-negative matrix where every row sums to 1 and every column sums to 1. Think of it as a matrix that redistributes signal across streams without amplifying the total magnitude, similar to a mixing board where the total volume going out always equals the total volume coming in. This constraint provides two critical stability guarantees. First, the spectral norm (the largest factor by which the matrix can stretch any vector) is bounded by 1. This means the transformation can never amplify signals, and it can only redistribute or shrink them. This prevents the numerical explosions that plagued unconstrained HC. Second, and more subtly, the set of doubly stochastic matrices is closed under multiplication. This means that if you multiply any number of doubly stochastic matrices together, the result is still doubly stochastic. This is what makes deep stacking stable because even after passing through 61 layers (as in DSV4-Pro), the cumulative effect of all residual transformations remains bounded and well-behaved. Further, the input and output mappings are also constrained. Both are passed through sigmoid functions to ensure all values are non-negative and bounded, preventing the possibility of signal cancellation where positive and negative contributions could destroy information. Thus, the output mapping includes a scaling factor of 2, allowing it to modestly amplify the layer's contribution when useful.

Rather than using fixed learned matrices, all three mappings are dynamically generated based on the current residual state. The residual state is flattened and normalized, then used to compute a dynamic (input-dependent) component for each mapping via learned projections. This dynamic component is combined with a static (input-independent) bias. Learnable gating factors, initialized to small values, control how much the dynamic component contributes. This initialization means that early in training, the model behaves like standard residual connections and gradually learns to utilize the expanded mixing capability as training progresses. For the residual transformation matrix specifically, the unconstrained output of the dynamic generation is projected onto the doubly stochastic constraint using the Sinkhorn-Knopp algorithm. This algorithm first exponentiates the matrix to ensure all entries are positive, then alternates between normalizing rows to sum to 1 and normalizing columns to sum to 1. After 20 iterations, the result converges to a valid doubly stochastic matrix that is as close as possible to the unconstrained output, preserving the model's expressive intent while enforcing the stability guarantee.

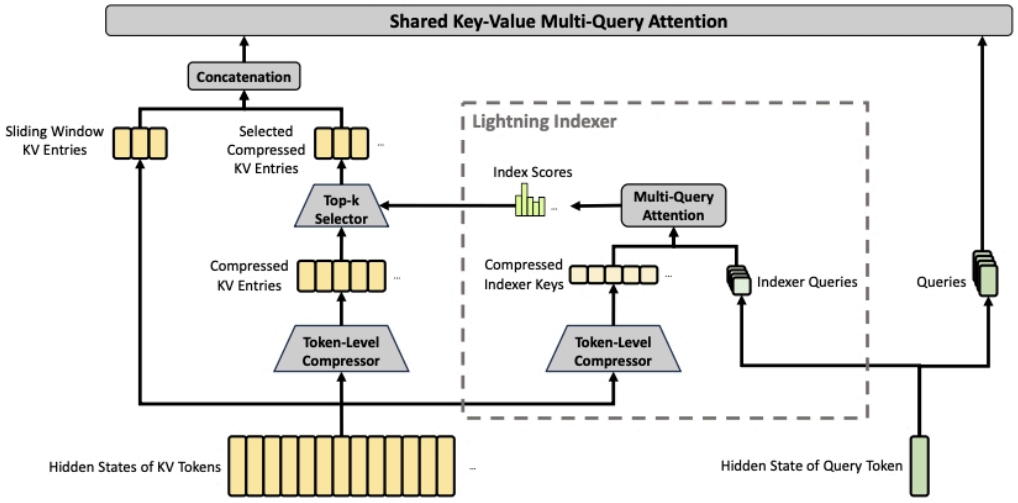
mHC increases activation memory consumption and communication volume between pipeline stages compared with conventional residual connections. DeepSeek mitigates these costs through three different strategies being fused kernels that avoid materializing intermediate tensors, a selective recomputation strategy that checkpoints only the minimum necessary tensors, and adjusted DualPipe scheduling that overlaps some mHC operations with other computation. The net wall-time overhead is constrained to 6.7% of the pipeline stage, which is a modest cost for the stability and expressivity gains that mHC provides.

Hybrid Attention with CSA and HCA

The most consequential architectural change in DSV4 is the replacement of Multi-Head Latent Attention (MLA) with a hybrid attention mechanism combining Compressed Sparse Attention (CSA) and Heavily Compressed Attention (HCA), used in an interleaved configuration across layers. This hybrid design is what enables DSV4 to support million-token contexts at practical cost. The fundamental insight here is that different layers can attend to context at different granularities. CSA provides fine-grained but sparse access as it lightly compresses the KV cache and then selects only the most relevant entries. HCA provides coarse but dense global visibility as it aggressively compresses the KV cache but lets every query see the entire compressed context. Interleaving these two mechanisms means the model alternates between a detailed-but-selective view and a complete-but-low-resolution view of the context, capturing both local detail and global structure.

CSA begins by compressing the KV cache along the sequence dimension. For every group of 4 consecutive tokens, their information is combined into a single compressed entry. The compression is not a simple average but rather it is a learned weighted combination where the model decides, for each dimension of each compressed entry, how much to draw from each of the constituent tokens. A distinctive design choice within the paper is the use of overlapping windows. Each compressed entry draws information from two sources, the 4 tokens in its own group and the 4 tokens from the preceding group, for a total of 8 contributing tokens. The model computes softmax weights (incorporating learned positional biases) over all 8 tokens jointly, then produces the compressed entry as a weighted sum. The overlap ensures that adjacent compressed blocks share information, preventing hard boundaries where context could be lost at block edges. Two separate series of KV entries and compression weights are computed from the input hidden states via four different learned projection matrices. This dual-series design allows the model to learn different representations for the "own group" contribution and the "neighboring group" contribution, giving it more flexibility in how it blends information across block boundaries.

Figure 2: Core Architecture of Compressed Sparse Attention (CSA)



Source: "DeepSeek-V4: Towards Highly Efficient Million-Token Context Intelligence", DeepSeek-AI

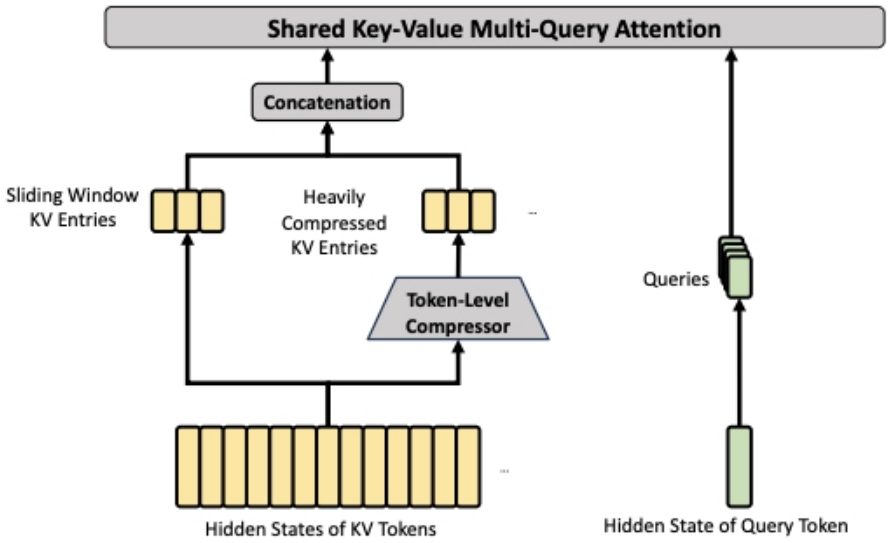
After compression reduces the sequence length by a factor of 4, CSA further reduces the computational cost by selecting only a subset of the compressed entries for each query token. This selection is performed by the Lightning Indexer, which is a lightweight attention mechanism that scores the relevance of each compressed block to each query. The indexer works through a two-stage projection. First, the query token's hidden state is down-projected to a compact latent vector (1024 dimensions for V4-Flash, 1536 for V4-Pro), which is shared with the main attention mechanism to avoid redundant computation. This latent vector is then up-projected to produce 64 indexer query heads. Separately, compressed indexer keys are produced through the same compression operation used for the main KV entries, but with their own projection matrices. The relevance score for each compressed block is computed as a weighted sum across all indexer heads. For each head, the dot product between the indexer query and the compressed indexer key is passed through a ReLU gate, meaning that only positively-correlated pairs contribute to the score, and anti-correlated pairs are zeroed out. Per-head weights, which are themselves learned projections from the query token, determine how much each head's opinion counts. The top-k blocks with the highest scores are then retained for core attention (V4-Flash selects 512 blocks; V4-Pro selects 1024).

A defining characteristic of DSV4's attention is that each compressed KV entry serves as both the attention key and the attention value. In standard attention, keys and values are separate projections that serve different roles, with keys determining where to attend, values determining what information to extract. By collapsing these into a single vector, DSV4 halves the per-entry cache size. This design choice trades some representational flexibility for a significant efficiency gain. Attention queries are produced from the same compressed latent vector used for the indexer (sharing the down-projection step), then up-projected to produce the full set of query heads (64 for V4-Flash, 128 for V4-Pro). Standard dot-product attention is then performed between these query heads and the selected compressed KV entries, in a Multi-Query configuration where all query heads share the same single set of KV entries.

The concatenated output of all attention heads is very large in DSV4's configuration, with up to 65,536 dimensions for DSV4-Pro. Directly projecting this down to the hidden dimension would be prohibitively expensive, so DSV4 introduces a two-stage grouped output projection, where heads are divided into groups (8 for Flash, 16 for Pro), each group's output is first projected to a compact 1024-dimensional intermediate, and then the concatenated intermediates are projected to the final hidden dimension. This reduces the output projection cost substantially while preserving the model's ability to combine information across heads.

HCA takes a fundamentally different approach, using extreme compression without sparsity. Where CSA compresses every 4 tokens into 1, HCA compresses every 128 tokens into 1, which is a 32x more aggressive ratio. After compression, every query attends to every compressed entry, and there is no indexer and no sparse selection. The result is that HCA gives each query a complete but very low-resolution view of the entire context. The compression mechanism is similar to CSA's but simpler, as each group of 128 tokens is compressed via a learned softmax-weighted combination with positional biases, but without overlapping windows, with each compression group draws only from its own tokens. The simpler design is feasible because HCA's primary role is global coverage rather than fine-grained detail. HCA uses the same shared key-value MQA and grouped output projection design as CSA. The query production mechanism is identical with low-rank compression of the hidden state followed by up-projection to query heads. CSA preserves fine-grained detail but uses sparsity, meaning each query only sees a subset of the context at high resolution, while HCA sacrifices spatial resolution but maintains global visibility, meaning each query sees the entire context at low resolution. Interleaving these mechanisms means the model alternates between these two perspectives, and the information flow through the residual stream (enhanced by mHC) allows each layer to benefit from the context gathered by previous layers using the other attention type.

Figure 3: Core Architecture of Heavily Compressed Attention (HCA)



Source: "DeepSeek-V4: Towards Highly Efficient Million-Token Context Intelligence", DeepSeek-AI

For both CSA and HCA, an RMSNorm operation is applied to each query head and each compressed KV entry immediately before the core attention operation. This normalization prevents attention logits from growing unboundedly, which is a known problem in deep Transformer training, and is particularly important for compatibility with the Muon optimizer, which can exacerbate attention logit explosion. RoPE is applied to only the last 64 dimensions of each query and KV entry vector. Because the KV entries serve as both keys and values, the attention output naturally carries absolute position information from the weighted sum of value vectors, which normally would break the model's ability to generalize across sequence positions. However, to compensate, RoPE with a negated position is also applied to the attention output, converting the absolute position information back into relative position information. The net effect is that the contribution of each KV entry to the output depends on its distance from the query, not its absolute position. Both CSA and HCA include a supplementary sliding window attention branch covering the most recent 128 uncompressed tokens, which first compensates for the blind spot created by compression, as a query cannot access uncompressed information from tokens within its own current compression block because compression only operates on complete blocks. And then second, recent tokens are typically the most informative for next-token prediction, and the sliding window ensures these are available at full resolution rather than only through compressed representations.

Each attention head has a learnable scalar that is added to the softmax denominator. This allows any head to reduce its total attention mass below 1, effectively providing a 'no-op' option when no KV entry is particularly relevant. When the model determines that attending to context would not help for a given query-head pair, it can route attention to this sink rather than being forced to distribute mass across irrelevant entries. The efficiency gains from hybrid CSA/HCA are compounded by several precision optimizations. A mixed-precision KV cache uses BF16 for the RoPE dimensions and FP8 for the remainder, nearly halving cache size. The Lightning Indexer's attention computation uses FP4 precision, and the index scores themselves are quantized from FP32 to BF16, achieving a 2x speedup for the top-k selector while maintaining 99.7% recall of relevant entries. Taking a standard GQA-8 attention configuration with BF16 precision as the baseline, DSV4's KV cache is reduced to approximately 2% of baseline at 1M context. Even compared with DSV3.2's already-efficient MLA, V4-Pro achieves 3.7x lower single-token FLOPs and 9.5x smaller KV cache at 1M tokens. DSV4-Flash pushes this to 9.8x lower FLOPs and 13.7x smaller cache. Additionally, the routed expert parameters use FP4 precision, which on future hardware optimized for mixed-format operations could achieve an additional 1/3 efficiency improvement.

Muon Optimizer

DSV4 also replaces AdamW with the Muon optimizer for most parameters, retaining AdamW only for embeddings, prediction heads, RMSNorm weights, and mHC static biases and gating factors. Muon's core operation is to take the accumulated gradient matrix and find its nearest orthogonal approximation via Newton-Schulz iterations, which converge toward the polar decomposition of the matrix. Intuitively, this removes the scaling information from the gradient update (which singular values encode) and preserves only the directional information (which the orthogonal factor encodes). The result is an update that adjusts the direction of each weight vector without disproportionately affecting well-conditioned versus ill-conditioned dimensions. DSV4's implementation uses a two-stage Newton-Schulz scheme with 10 iterations. The first 8 iterations use aggressive coefficients that rapidly drive the singular values toward 1 but may overshoot slightly. The final 2 iterations use conservative coefficients that precisely stabilize the singular values at 1. Additionally, a key engineering challenge is compatibility with ZeRO parallelism, which ordinarily shards parameters across ranks, but Muon needs the full gradient matrix on a single rank for orthogonalization. DeepSeek solves this with a hybrid assignment strategy involving a knapsack algorithm assigns complete parameter matrices to ranks without splitting any matrix, with padding (typically under 10% overhead) to equalize bucket sizes. For MoE parameters, all expert projections of the same type are flattened across layers and distributed without splitting. Furthermore, an important discovery is that Newton-Schulz iterations remain stable in BF16, enabling gradient communication at half precision and halving communication volume.

Figure 4: Muon Optimizer for DeepSeek-V4

Algorithm 1 Muon Optimizer for DeepSeek-V4

Require: Learning rate η, momentum μ, weight decay λ, update rescaling factor γ

for each training step t **do**
  for each logically independent weight $W \in \mathbb{R}^{n \times m}$ **do**
    $G_t = \nabla_W \mathcal{L}_t(W_{t-1})$ ▷ Compute gradients
    $M_t = \mu M_{t-1} + G_t$ ▷ Accumulate momentum buffer
    $O'_t = \text{HybridNewtonSchulz}(\mu M_t + G_t)$ ▷ Nesterov trick and hybrid Newton-Schulz
    $O_t = O'_t \cdot \sqrt{\max(n, m)} \cdot \gamma$ ▷ Rescale the update RMS
    $W_t = W_{t-1} \cdot (1 - \eta\lambda) - \eta O_t$ ▷ Perform weight decay and update
  end for
end for

Source: "DeepSeek-V4: Towards Highly Efficient Million-Token Context Intelligence", DeepSeek-AI

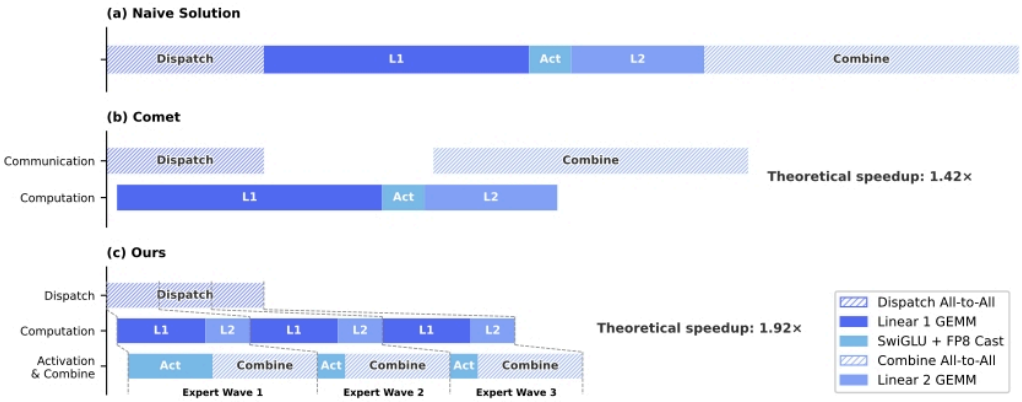
General Infrastructure

The DSV4 paper describes six major infrastructure contributions, each addressing a specific bottleneck in training or inference at scale.

Fine-Grained Communication-Computation Overlap in Expert Parallelism

Expert Parallelism (EP) distributes different experts across different devices, requiring all-to-all communication to dispatch tokens to their assigned experts and collect results, with this communication overhead having become the dominant scaling bottleneck for MoE models. Each MoE layer decomposes into four stages that are made up of two communication-bound (Dispatch and Combine all-to-all operations) and two computation-bound (the two GEMM operations in each expert's feed-forward network) stages. DeepSeek's key insight is that within a single MoE layer, the total communication time is less than the total computation time, which means that if communication and computation can be fully overlapped, computation becomes the sole bottleneck and communication cost effectively disappears.

Figure 5: DeepSeek EP Scheme



Source: "DeepSeek-V4: Towards Highly Efficient Million-Token Context Intelligence", DeepSeek-AI

To achieve this overlap, DSV4 partitions experts into small groups called waves. As soon as all experts within a wave have finished receiving their tokens, computation on that wave begins immediately without waiting for the remaining experts to complete their communication. In steady state, three operations proceed concurrently including computation on the current wave, data transfer for the next wave, and result transmission from completed waves. This creates a fine-grained pipeline that keeps both compute units and network links continuously active. Prior work (such as Comet) achieves coarser overlap by pairing Dispatch with Linear-1 and Linear-2 with Combine as two separate windows. DSV4's wave-based approach achieves a theoretical 1.92x speedup (versus Comet's 1.42x for the V4-Flash configuration) by enabling all four stages to overlap simultaneously across different waves. In practice, the approach achieves 1.50-1.73x speedup for general inference and up to 1.96x for latency-sensitive RL rollouts.

The paper provides a precise threshold for when full overlap is achievable. For DSV4-Pro, each token-expert pair requires 6 times h times d FLOPs (for SwiGLU gate, up, and down projections) but only 3 times h bytes of communication (FP8 Dispatch plus BF16 Combine). Full overlap is possible as long as the ratio of peak compute throughput to interconnect bandwidth does not exceed 2 times d, which equals 6,144 FLOPs per byte. In concrete terms, each GBps of interconnect bandwidth can hide 6.1 TFLOP/s of compute. Once bandwidth meets this threshold, additional bandwidth investment yields diminishing returns. This analysis provides a concrete framework for future hardware design as rather than scaling bandwidth unconditionally, hardware should target the balance point defined by the model's compute-communication ratio. The paper also notes that extreme kernel fusion drives compute, memory, and network to simultaneous high load, making power throttling a key performance limiter which suggests that future hardware should provide sufficient power headroom for fully concurrent workloads.

Flexible and Efficient Kernel Development with TileLang

DSV4's elaborate architecture would have required hundreds of individual operators if implemented naively, thus DeepSeek adopts TileLang, a Domain-Specific Language built as an extension of TVM, to develop fused kernels that replace the vast majority of these operators. DeepSeek makes a few innovations within TileLang that make it particularly effective, starting with HostHost Codegen that eliminates CPU-side orchestration overhead. Rather than performing runtime contract checks in Python (which incur tens to hundreds of microseconds per invocation), TileLang co-generates a lightweight host launcher alongside each device kernel at the IR level. This launcher embeds all necessary metadata such as data types, shape constraints, layout assumptions, and performs validation and argument marshaling in compiled code rather than interpreted Python. The result is per-invocation overhead of less than one microsecond.

Second, an SMT-solver integration provides formal integer analysis for tensor programs. TileLang integrates the Z3 theorem prover into its algebraic system, translating tensor index expressions into quantifier-free non-linear integer arithmetic, which enables the compiler to formally verify integer properties needed for optimizations such as vectorization, barrier insertion, and code simplification, including advanced cases like vectorization over variable tensor shapes that simpler analysis frameworks cannot handle. The compilation time overhead is then restricted to a few seconds. Third, numerical precision and bitwise reproducibility are prioritized by default. Fast-math optimizations are disabled at the compiler level. Precision-affecting approximations are available only as explicit opt-in operators. IEEE-compliant intrinsics with explicit rounding modes are provided for strict numerical correctness. Layout annotations allow developers to pin down accumulation and evaluation order, enabling bit-identical outputs when validating against reference CUDA implementations.

Batch-Invariant and Deterministic Kernel Libraries

DSV4 achieves end-to-end bitwise batch-invariant and deterministic training, meaning the output of any given token is identical regardless of batch position, and identical inputs always produce identical outputs across runs. These properties are critical for debugging (especially isolating the cause of loss spikes), stability analysis, and ensuring consistent behavior across training, post-training, and inference pipelines.

Achieving batch invariance requires solving several problems that arise from standard high-performance computing techniques. The most significant is attention computation, with the standard approach to high-throughput attention (split-KV) distributing a single sequence's attention computation across multiple Stream Multiprocessors (SMs) for load balancing. However, different batch positions lead to different SM assignments, producing different floating-point accumulation orders and thus different numerical results. DeepSeek develops a dual-kernel strategy, with the primary kernel computing attention for an entire sequence within a single SM, guaranteeing batch invariance through deterministic accumulation order. However, using a single SM per sequence creates severe wave-quantization problems as the last wave of SMs is often partially filled, leading to poor utilization. The secondary kernel addresses this by using multiple SMs for single sequences in the final wave, with accumulation order carefully designed to exactly match the primary kernel. Cross-SM data exchange is handled via distributed shared memory within thread-block clusters. The result is negligible overhead for batch invariance. For matrix multiplication, the standard cuBLAS library is replaced end-to-end with DeepGEMM, which provides batch-invariant guarantees. Standard split-k techniques for small batch sizes are abandoned in favor of custom optimizations that match or exceed split-k performance without sacrificing invariance.

Deterministic training eliminates non-deterministic accumulation order arising from atomic addition instructions, primarily in the backward pass. Three specific solutions are implemented. For sparse attention backward, where gradients for KV tokens would normally use atomicAdd (which produces non-deterministic results due to floating-point non-associativity), DSV4 allocates separate accumulation buffers per SM and performs a global deterministic summation afterward. For MoE backward, where concurrent multi-rank writes create non-deterministic buffer positions, a token order pre-processing mechanism within each rank combined with cross-rank buffer isolation ensures deterministic accumulation. For mHC's small-dimension matrix multiplication (output dimension of only 24), where split-k is required for performance, each split part is output separately and reduced deterministically in a subsequent kernel.

FP4 Quantization-Aware Training

DSV4 applies MXFP4 quantization-aware training during post-training to two components, the MoE expert weights and the query-key path in the CSA indexer. For expert weights, the optimizer's FP32 master weights are quantized to FP4 and then dequantized back to FP8 for computation. A critical insight makes this pipeline efficient, being that the dequantization from FP4 to FP8 is lossless. FP8 in E4M3 format has 2 additional exponent bits compared to FP4's E2M1 format, providing enough dynamic range to fully absorb the fine-grained scale factors used in FP4 quantization. This means the entire QAT pipeline can reuse the existing FP8 training framework without modification, thus gradients flow through the FP8 weights and back to the FP32 master weights via the Straight-Through Estimator. For the CSA indexer's query-key path, activations are cached, loaded, and multiplied entirely in FP4, accelerating sparse selection under long contexts. The index scores are also quantized from FP32 to BF16, yielding a 2x speedup for the top-k selector with only 0.3% recall loss. During inference and RL rollouts (which do not involve backward passes), real FP4 weights are used directly rather than simulated quantization, ensuring behavioral consistency between training sampling and online deployment.

Training Framework

Muon requires access to the full gradient matrix for orthogonalization, conflicting with ZeRO's parameter sharding. DeepSeek's solution uses a knapsack algorithm to assign complete parameter matrices to ZeRO ranks without splitting any logically independent matrix, padding buckets (under 10% overhead) for efficient communication. For MoE parameters, expert projections of the same type are flattened across layers and distributed without splitting. Consecutive parameters of identical shape are automatically merged for batched Newton-Schulz execution. BF16 gradient communication (exploiting the stability of Newton-Schulz in BF16) halves bandwidth requirements, with a two-phase reduce (all-to-all exchange followed by local FP32 summation) maintaining numerical robustness.

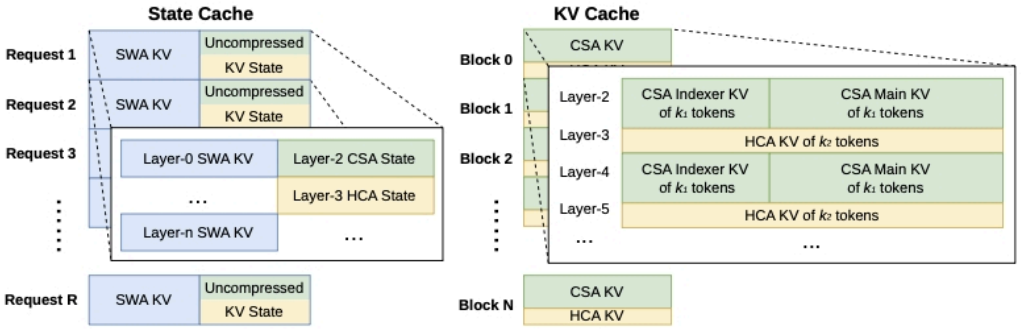
Furthermore, context parallelism partitions the sequence across ranks, but compression requires groups of consecutive tokens that may straddle rank boundaries. DSV4 handles this with a two-stage approach, where first, each rank sends its last m uncompressed KV entries to the next rank, enabling boundary-crossing compression. Second, an all-gather collects locally compressed entries across all ranks, and a fused operator reorganizes them into the complete compressed KV sequence with padding at the tail.

Finally, rather than deciding at module granularity whether to retain or recompute activations, DSV4 allows developers to annotate individual tensors for checkpointing. The framework traces the computation graph with TorchFX, identifies the minimal subgraph needed to recompute each annotated tensor, and inserts this recomputation into the backward pass. The implementation frees GPU memory for annotated tensors and reuses storage pointers from recomputed tensors (no memory copy), with automatic deduplication for tensors sharing storage. This achieves the fine-grained control of manual implementation with the convenience of automatic differentiation and zero additional overhead.

Inference Framework

The hybrid attention mechanism creates KV entries with different sizes and update rules across layers. DSV4 organizes the cache into two components. The Classical KV Cache stores compressed CSA and HCA entries, allocated in blocks covering 128 original tokens (the least common multiple of CSA's compression rate 4 and HCA's compression rate 128), producing 32 CSA compressed tokens and 1 HCA compressed token per block. The State Cache stores sliding window entries and uncompressed tail tokens not yet ready for compression, allocated as fixed-size per-request blocks from a pre-allocated pool. This separation avoids the assumptions of PagedAttention (uniform cache sizes, uniform eviction policies) that the hybrid architecture violates. A co-designed sparse attention kernel accommodates variable compressed token counts across layers without performance degradation.

Figure 6: KV Cache Layout for DeepSeek-V4



Source: "DeepSeek-V4: Towards Highly Efficient Million-Token Context Intelligence", DeepSeek-AI

For shared-prefix requests, compressed entries are stored to disk and reused on cache hits. For SWA entries (approximately 8x larger than compressed entries), three strategies trade off storage against recomputation. Full Caching stores everything but creates write-intensive access patterns on SSDs. Periodic Checkpointing stores snapshots every p tokens, recomputing the tail on hit. Zero Caching stores nothing and recomputes the last n_win times L tokens (where L is the layer count) from cached compressed entries, which is feasible because each layer's SWA entry depends only on the previous layer's most recent n_win tokens.

Potential Risks

Required Disclosures

Best-of-Breed: Expected to outperform on a risk adjusted basis over a five-year time horizon, but may be fully valued over a 12-18 month time horizon.

D.A. Davidson & Co. is serving as capital markets advisor to Willow Lane Acquisition Corp. in the SPAC merger with Boost Run, announced on or about September 16, 2025. Per the terms of this advisory agreement, D.A. Davidson & Co. will receive a fee upon the completion of the merger.

D.A. Davidson & Co. makes a market in Apple Inc., Adobe Inc., Advanced Micro Devices, Inc., Amazon.com, Inc., Broadcom Inc., Salesforce.com, Inc., CoreWeave, Inc., Datadog, Inc., Alphabet Inc., Intel Corporation, MongoDB, Inc., Meta Platforms, Inc., Microsoft Corporation, Nebius Group N.V., NVIDIA Corporation, Oracle Corporation, Snowflake Inc. and Boost Run LLC.

D.A. Davidson & Co, or any of its affiliates, does or seeks to do business with companies covered in its research reports. As a result, investors should be aware that the firm may have a conflict of interest that could affect the objectivity of this report. Investors should consider this report as only a single factor in making their investment decision.

D.A. Davidson & Co. is a full service investment firm that provides both brokerage and investment banking services. Alexander Platt, the research analyst principally responsible for the preparation of this report has received and is eligible to receive compensation, including bonus compensation, based on D.A. Davidson's overall operating revenues, including revenues generated by its investment banking and institutional equities activities. D.A. Davidson & Co.'s analysts, however, are not directly compensated for involvement in specific investment banking transactions.
I, Alexander Platt, attest that (i) all the views expressed in this research report accurately reflect my personal views about the common stock of the subject company, and (ii) no part of my compensation was, is, or will be, directly or indirectly, related to the specific recommendations or views expressed in this report.

Rating Information
D.A. Davidson & Co.'s Institutional Research Rating Scale Definitions (maintained since October 10, 2017); information regarding our previous definitions is available upon request:
BUY: Expected to produce a total return of over 15% on a risk adjusted basis over the next 12-18 months
NEUTRAL: Expected to produce a total return of -15% to +15% on a risk adjusted basis over the next 12-18 months
UNDERPERFORM: Expected to lose value of over 15% on a risk adjusted basis over the next 12-18 months

Rating Distribution (as of 3/31/26)	Coverage Universe Distribution			Investment Banking Distribution		
	IR	WMR	Combined	IR	WMR	Combined
BUY (Buy)	57%	85%	61%	9%	0%	9%
NEUTRAL (Hold)	42%	15%	39%	4%	0%	3%
UNDERPERFORM (Sell)	1%	0%	0%	0%	0%	0%

IR denotes Institutional Research; WMR denotes Wealth Management Research whose rating scale is Buy/Add, Neutral, Sell/Reduce. Investment Banking Distribution denotes companies from whom D.A. Davidson & Co. has received compensation in the last 12 months. Best-of-Breed: Expected to outperform on a risk adjusted basis over a five-year time horizon.

Target prices are our Institutional Research Department's evaluation of price potential over the next 12 months, based upon our assessment of future earnings and cash flow, comparable company valuations, growth prospects and other financial criteria. Certain risks may impede achievement of these price targets including, but not limited to, broader market and macroeconomic fluctuations and unforeseen changes in the subject company's fundamentals or business trends.
While the Best-of-Breed designation does not contain a separate rating and/or price target from that of the standard ratings system referenced above, the expectation is that the security, based on the 12 criteria utilized in assessing the "Best-of-Breed" designation, will outperform over a five-year time horizon, not the standard 12-18 month time horizon.

For a copy of the most recent reports containing all required disclosure information for covered companies referenced in this report, please contact your D.A. Davidson & Co. representative or call 1-800-755-7848.

Other Disclosures

Information contained herein has been obtained by sources we consider reliable, but is not guaranteed and we are not soliciting any action based upon it. Any opinions expressed are based on our interpretation of data available to us at the time of the original publication of the report. These opinions are subject to change at any time without notice. Investors must bear in mind that inherent in investments are the risks of fluctuating prices and the uncertainties of dividends, rates of return and yield. Investors should also remember that past performance is not necessarily an indicator of future performance and D.A. Davidson & Co. makes no guarantee, express or implied, as to future performance. Investors should note this report was prepared by D.A. Davidson & Co.'s Institutional Research Department for distribution to D.A. Davidson & Co.'s institutional investor clients and assumes a certain level of investment sophistication on the part of the recipient. Readers, who are not institutional investors or other market professionals, should seek the advice of their individual investment advisor for an explanation of this report's contents, and should always seek such advisor's advice before making any investment decisions. Consensus estimates are obtained from Capital IQ. Further information and elaboration will be furnished upon request.

Other Companies Mentioned in this Report

Company Name	Ticker	Rating	Price
Amazon.com, Inc.	AMZN	NEUTRAL	$255.08
Apple Inc.	AAPL	NEUTRAL	$273.43
Alphabet Inc.	GOOGL	NEUTRAL	$338.89
Meta Platforms, Inc.	META	BUY	$659.15
Microsoft Corporation	MSFT	BUY	$415.75
NVIDIA Corporation	NVDA	BUY	$199.64
Advanced Micro Devices, Inc.	AMD	BUY	305.33
Intel Corporation	INTC	NEUTRAL	66.78
Taiwan Semiconductor Manufacturing Co.	TSM	BUY	382.66
Broadcom Inc.	AVGO	NEUTRAL	$419.94
CoreWeave, Inc.	CRWV	BUY	$117.42
Nebius Group N.V.	NBIS	BUY	$157.08
Oracle Corporation	ORCL	BUY	$176.28
Boost Run LLC	WLAC	BUY	15.00
Snowflake Inc.	SNOW	BUY	$146.40
Datadog, Inc.	DDOG	BUY	$127.86
MongoDB, Inc.	MDB	BUY	$258.11
Salesforce.com, Inc.	CRM	NEUTRAL	$173.30
Adobe Inc.	ADBE	BUY	$238.98