DeepSeek-V3 Technical Report

Technical Report Overview

Ion Lipsiuc

lipsiuci@tcd.ie

February 26, 2025

Introduction

- DeepSeek-V3 is a large Mixture of Experts (MoE) language model with 671 billion total parameters, of which 37 billion are activated per token.
- Key innovations:
  - Multi-Head Latent Attention (MLA) for efficient inference.
  - DeepSeekMoE architecture for cost-effective training.
  - Auxiliary-Loss-Free Load Balancing Strategy.
  - Multi-Token Prediction (MTP) training objective.
- Pre-trained on 14.8 trillion tokens, achieving state-of-the-art performance.
- Training completed in 2.788 million H800 GPU hours, demonstrating remarkable stability.

Multi-Head Latent Attention

- MLA reduces Key-Value (KV) cache during inference by compressing keys and values.
- Low-rank joint compression for attention keys and values.
- Formulation:

$$\mathbf{k}_{t,i} = \left[\mathbf{k}_{t,i}^{C}; \mathbf{k}_{t}^{R}\right], \quad \mathbf{v}_{t}^{C} = W^{UV}\mathbf{c}_{t}^{KV},$$

  where:
  - $\mathbf{k}_{t,i}^{C}$: Compressed latent vector for keys.
  - $\mathbf{k}_{t}^{R}$: Decoupled key with Rotary Positional Embedding (RoPE).
  - $\mathbf{v}_{t}^{C}$: Compressed latent vector for values.
  - $\mathbf{c}_{t}^{KV}$: Latent representation used for both keys and values.
  - W^{UV}: A learnable projection matrix.
- Only $\mathbf{c}_{t}^{KV}$ and $\mathbf{k}_{t}^{R}$ are cached, reducing memory usage significantly.

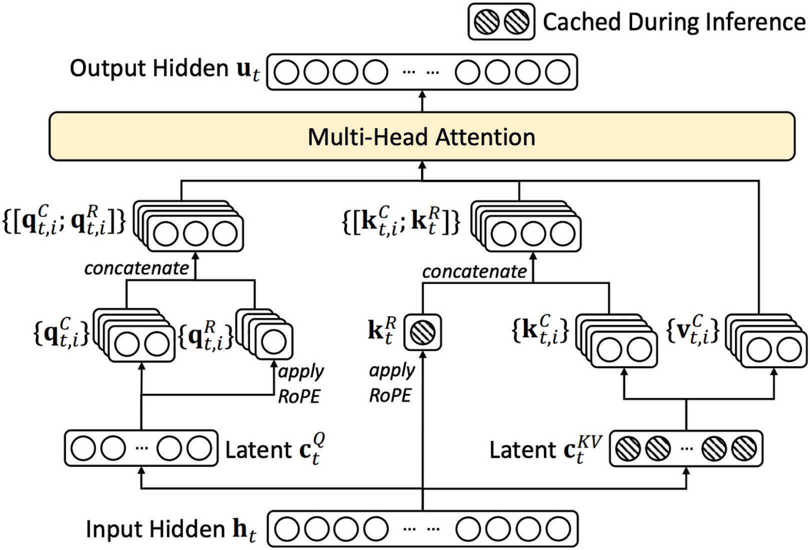

Figure: Keys and values are compressed into latent representations, with a small set of components cached during inference to reduce memory usage.

Mixture of Experts

- DeepSeekMoE uses a combination of shared experts and routed experts for efficiency.
- Formulation:

$$\mathbf{h}'_t = \mathbf{u}_t + \sum_{i=1}^{N_s} \mathrm{FFN}_i^{(s)}(\mathbf{u}_t) + \sum_{i=1}^{N_r} g_{i,t} \mathrm{FFN}_i^{(r)(\mathbf{u}_t)},$$

  where:
  - N_s: Number of shared experts (applied to every token).
  - N_r: Number of routed experts (selectively applied to tokens).
  - $g_{i,t}$: Gating value for expert i at time step t.
- Auxiliary-loss-free load balancing ensures balanced expert usage without adding extra losses:

$$g'_{i,t} = \begin{cases} s_{i,t}, & \text{if } s_{i,t} + b_i \in \mathrm{Topk}(\{s_{j,t} + b_j \mid 1 \leq j \leq N_r\}, K_r), \\ 0, & \text{otherwise.} \end{cases}$$

  Here $s_{i,t}$ is a raw score for expert i and b_i is a learned bias term.

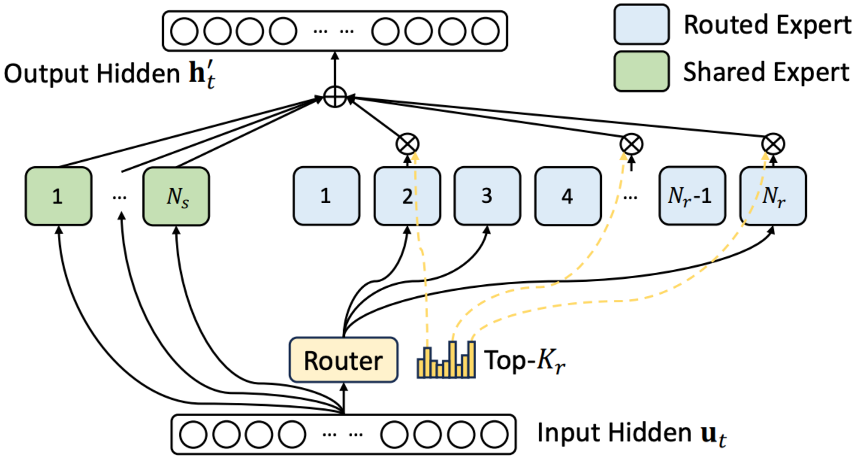

Figure: The router selects which experts (routed experts) to activate per token. Shared experts are always applied.

Multi-Token Prediction

- MTP extends the prediction scope to multiple future tokens at once.
- Each module in MTP predicts one of the next several tokens, allowing partial parallelism.
- Formulation:

$$\mathbf{h}_i^k = M_k[\mathrm{RMSNorm}(\mathbf{h}_i^{k-1}); \mathrm{RMSNorm}(\mathrm{Emb}(t_{i+k}))],$$

  where:
  - $\mathbf{h}_i^{k-1}$: Representation from the $(k-1)^{\text{th}}$ module.
  - $\mathrm{Emb}(t_{i+k})$: Embedding of the $(i+k)^{\text{th}}$ token.
  - M_k: The k^{th} MTP module (e.g., a Transformer block).
- Overall training objective:

$$\mathcal{L}_{\text{MTP}} = \frac{\lambda}{D} \sum_{k=1}^{D} \mathcal{L}_{\text{MTP}}^k,$$

  where $\mathcal{L}_{\text{MTP}}^k$ is the cross-entropy loss at depth k, and D is the number of MTP modules.

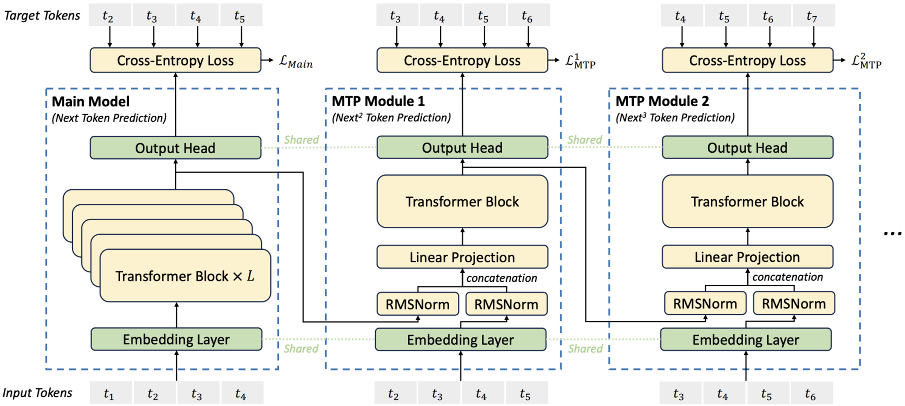

Figure: Several modules predict future tokens in parallel, each focusing on a different next token.

Compute Cluster and Training Framework

- Trained on a cluster with **2048 NVIDIA H800 GPUs**.
- **DualPipe** algorithm for efficient pipeline parallelism:
  - Overlaps computation and communication.
  - Reduces pipeline bubbles and communication overhead.
- **DualPipe scheduling**:
  - Divides chunks into attention, all-to-all dispatch, MLP, and all-to-all combine steps.
  - Overlaps forward and backward computation-communication phases.
- **Efficient cross-node all-to-all communication**:
  - Utilizes InfiniBand (IB) and NVLink bandwidths.
  - Limits tokens to four nodes to reduce IB traffic.

Figure: Example of pipeline stages (attention, MLP, dispatch/collect steps) overlapping forward (colored blocks) and backward (triangles) phases.

Figure: This timeline illustrates how forward and backward passes are distributed across eight devices, with color-coded segments indicating computation or communication tasks.

FP8 Training

- **FP8 mixed precision training framework**:
  - Most compute-intensive operations in FP8 for speed and reduced memory.
  - Key operations in higher precision (BF16 or FP32) to maintain numerical stability.
- **Fine-grained quantization**:
  - Activations: 1x128 tile basis (small sub-blocks).
  - Weights: 128x128 block basis.
- **Low-precision storage and communication**:
  - Activations cached in FP8 for the backward pass.
  - Optimizer states stored in BF16 to reduce memory usage.
- Achieves high training efficiency with minimal precision loss.

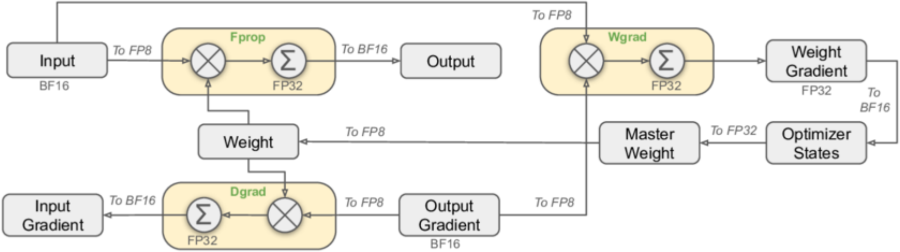

Figure: The forward pass (Fprop) and backward pass (Dgrad, Wgrad) are done in FP8. A master copy of the weights is kept in BF16 or FP32 for numerical stability.

Pre-Training

- Pre-trained on **14.8 trillion** high-quality tokens.
- **Two-stage context length extension**:
  - Stage 1: Extend context length to 32K.
  - Stage 2: Extend context length to 128K.
- Remarkably stable training process:
  - No irrecoverable loss spikes or rollbacks.
- Post-training includes:
  - **Supervised Fine-Tuning (SFT)**.
  - **Reinforcement Learning (RL)**.
  - **Distillation** from DeepSeek-R1 for better reasoning.

Conclusion

- DeepSeek-V3 is a state-of-the-art open-source MoE model.
- Key innovations: **MLA**, **DeepSeekMoE**, **MTP**, and **FP8** training.
- Achieves high performance with economical training costs.