本文由 AINLP 公众号整理翻译，更多 LLM 资源请扫码关注!

DeepSeek-V3.2-Exp: Boosting Long-Context Efficiency with DeepSeek Sparse Attention

DeepSeek-AI

research@deepseek.com

Abstract

We introduce DeepSeek-V3.2-Exp, an experimental sparse-attention model, which equips DeepSeek-V3.1-Terminus with DeepSeek Sparse Attention (DSA) through continued training. With DSA, a fine-grained sparse attention mechanism powered by a lightning indexer, DeepSeek-V3.2-Exp achieves significant efficiency improvements in both training and inference, especially in long-context scenarios. The model checkpoints are available at https://huggingface.co/deepseek-ai/DeepSeek-V3.2-Exp.

1. Architecture

Compared with DeepSeek-V3.1-Terminus, the last version of DeepSeek-V3.1, the only architectural modification of DeepSeek-V3.2-Exp is the introduction of DeepSeek Sparse Attention (DSA) through continued training.

Prototype of DSA. The prototype of DSA primarily consists of two components: a lightning indexer and a fine-grained token selection mechanism.

The **lightning indexer** computes the index score $I_{t,s}$ between the query token $\mathbf{h}_t \in \mathbb{R}^d$ and a preceding token $\mathbf{h}_s \in \mathbb{R}^d$, determining which tokens to be selected by the query token:

$$I_{t,s} = \sum_{j=1}^{H^I} w_{t,j}^I \cdot \mathrm{ReLU}\left(\mathbf{q}_{t,j}^I \cdot \mathbf{k}_s^I\right), \tag{1}$$

where H^I denotes the number of indexer heads; $\mathbf{q}_{t,j}^I \in \mathbb{R}^{d^I}$ and $w_{t,j}^I \in \mathbb{R}$ are derived from the query token $\mathbf{h}_t$; and $\mathbf{k}_s^I \in \mathbb{R}^{d^I}$ is derived from the preceding token $\mathbf{h}_s$. We choose ReLU as the activation function for throughput consideration. Given that the lightning indexer has a small number of heads and can be implemented in FP8, its computational efficiency is remarkable.

Given the index scores $\{I_{t,s}\}$ for each query token $\mathbf{h}_t$, our **fine-grained token selection mechanism** retrieves only the key-value entries $\{\mathbf{c}_s\}$ corresponding to the top-k index scores. Then, the attention output $\mathbf{u}_t$ is computed by applying the attention mechanism between the query token $\mathbf{h}_t$ and the sparsely selected key-value entries $\{\mathbf{c}_s\}$:

$$\mathbf{u}_t = \mathrm{Attn}\left(\mathbf{h}_t, \left\{\mathbf{c}_s \,\middle|\, I_{t,s} \in \text{Top-k}(I_{t,:})\right\}\right). \tag{2}$$

DeepSeek-V3.2-Exp：通过DeepSeek稀疏注意力提升长上下文效率

深度求索人工智能

research@deepseek.com

摘要

我们推出DeepSeek-V3.2-Exp实验性稀疏注意力模型，该模型通过持续训练为DeepSeek-V3.1-Terminus装备了DeepSeek稀疏注意力（DSA）机制。借助由闪电索引器驱动的细粒度稀疏注意力机制，DeepSeek-V3.2-Exp在训练和推理效率上实现显著提升，尤其在长上下文场景中表现突出。模型检查点可在https://huggingface.co/deepseek-ai/DeepSeek-V3.2-Exp获取。

1. 架构

与DeepSeek-V3.1-Terminus（即DeepSeek-V3.1的最终版本）相比，DeepSeek-V3.2-Exp在架构上的唯一改动是通过持续训练引入了DeepSeek稀疏注意力（DSA）。

DSA原型。DSA的原型主要由两个组件构成：一个闪电索引器和一个细粒度令牌选择机制。

闪电索引器计算查询标记$\mathbf{h}_t \in \mathbb{R}^d$与前置标记$\mathbf{h}_s \in \mathbb{R}^d$之间的索引分数$I_{t,s}$，确定哪些标记将被查询标记选中：

$$I_{t,s} = \sum_{j=1}^{H^I} w_{t,j}^I \cdot \mathrm{ReLU}\left(\mathbf{q}_{t,j}^I \cdot \mathbf{k}_s^I\right), \tag{1}$$

其中H^I表示索引器头数；$\mathbf{q}_{t,j}^I \in \mathbb{R}^{d^I}$和$w_{t,j}^I \in \mathbb{R}$源自查询标记$\mathbf{h}_t$；$\mathbf{k}_s^I \in \mathbb{R}^{d^I}$源自前一个标记$\mathbf{h}_s$。出于吞吐量考虑，我们选择ReLU作为激活函数。鉴于闪电索引器头数较少且可采用FP8实现，其计算效率非常显著。

给定每个查询标记$\mathbf{h}_t$的索引分数$\{I_{t,s}\}$，我们的细粒度标记选择机制仅检索与最高k个索引分数对应的键值条目$\{\mathbf{c}_s\}$。随后，通过在查询标记$\mathbf{h}_t$与稀疏选定的键值条目$\{\mathbf{c}_s\}$之间应用注意力机制，计算出注意力输出$\mathbf{u}_t$：

$$\mathbf{u}_t = \mathrm{Attn}\left(\mathbf{h}_t, \left\{\mathbf{c}_s \,\middle|\, I_{t,s} \in \text{Top-k}(I_{t,:})\right\}\right). \tag{2}$$

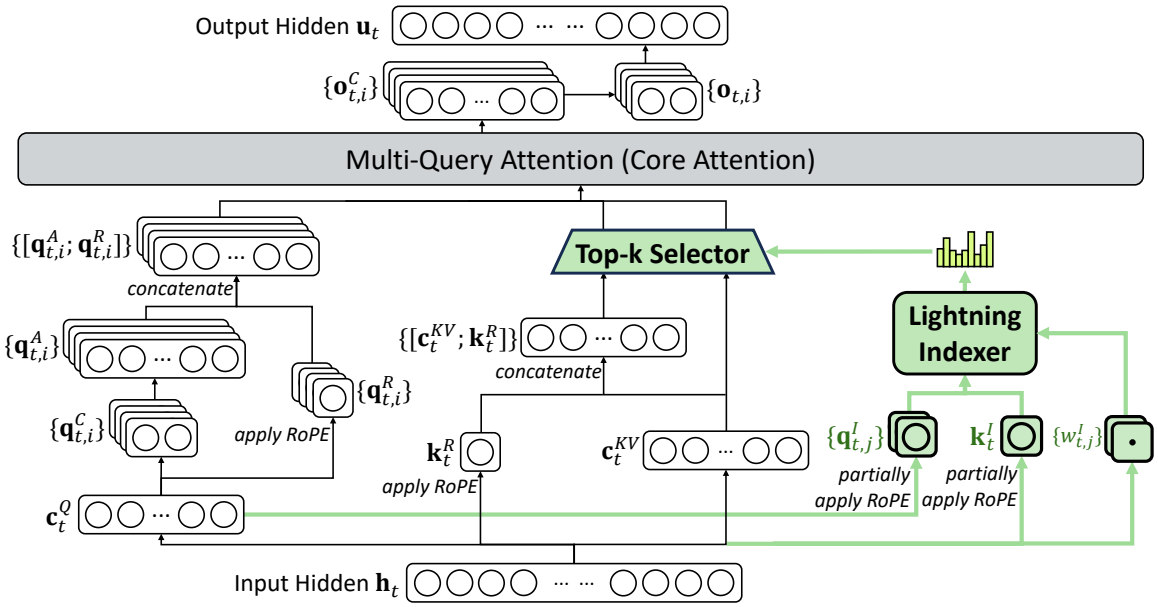


Figure 1 | Attention architecture of DeepSeek-V3.2-Exp, where DSA is instantiated under MLA. The green part illustrates how DSA selects the top-k key-value entries according to the indexer.

Instantiate DSA Under MLA. For the consideration of continued training from DeepSeek-V3.1-Terminus, we instantiate DSA based on MLA (DeepSeek-AI, 2024) for DeepSeek-V3.2-Exp. At the kernel level, each key-value entry must be shared across multiple queries for computational efficiency (Yuan et al., 2025). Therefore, we implement DSA based on the MQA (Shazeer, 2019) mode of MLA[1], where each latent vector (the key-value entry of MLA) will be shared across all query heads of the query token. The DSA architecture based on MLA is illustrated in Figure 1. We also provide an open-source implementation of DeepSeek-V3.2-Exp[2] to specify the details unambiguously.

2. Training

Starting from a base checkpoint of DeepSeek-V3.1-Terminus, whose context length has been extended to 128K, we perform continued pre-training followed by post-training to create DeepSeek-V3.2-Exp.

2.1. Continued Pre-Training

The continued pre-training of DeepSeek-V3.2-Exp consists of two training stages. For both stages, the distribution of training data is totally aligned with the 128K long context extension data used for DeepSeek-V3.1-Terminus.

Dense Warm-up Stage. We first use a short warm-up stage to initialize the lightning indexer. In this stage, we keep dense attention and freeze all model parameters except for the lightning indexer. To align the indexer outputs with the main attention distribution, for the t-th query token, we first aggregate the main attention scores by summing across all attention heads.

[1] We illustrate the difference between the MQA and MHA modes of MLA in Appendix A.

[2] https://huggingface.co/deepseek-ai/DeepSeek-V3.2-Exp/tree/main/inference

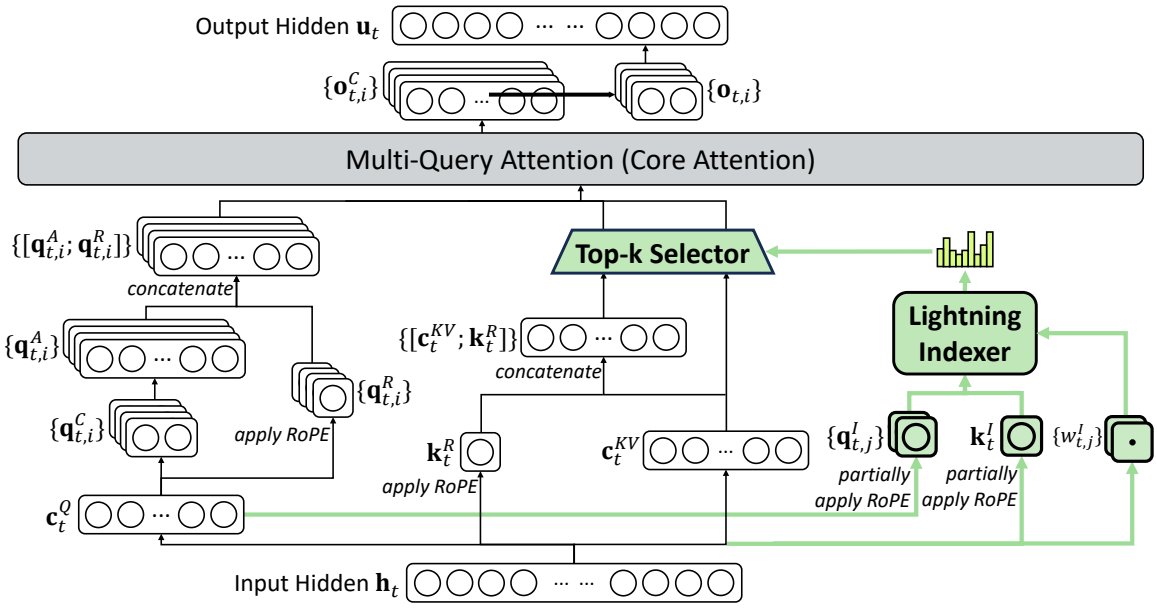


图1 | DeepSeek-V3.2-Exp的注意力架构，其中DSA在MLA框架下实例化。绿色部分展示了DSA如何根据索引器筛选前k个键值条目。

在MLA框架下实例化DSA。考虑到从DeepSeek-V3.1-Terminus进行持续训练的需求，我们基于MLA（DeepSeek-AI，2024）为DeepSeek-V3.2-Exp实例化了DSA。在核心层面，为提升计算效率，每个键值条目必须跨多个查询共享（Yuan等人，2025）。因此，我们基于MLA[1]的MQA模式（Shazeer，2019）实现DSA，其中每个潜在向量（MLA的键值条目）将在查询令牌的所有查询头之间共享。基于MLA的DSA架构如图1所示。我们还开源了DeepSeek-V3.2-Exp[2]的实现，以明确具体细节。

2. 训练

从基础检查点DeepSeek-V3.1-Terminus出发（其上下文长度已扩展至128K），我们通过持续预训练与后训练流程，最终构建出DeepSeek-V3.2-Exp模型。

2.1. 持续预训练

DeepSeek-V3.2-Exp的持续预训练包含两个训练阶段。在这两个阶段中，训练数据的分布与用于DeepSeek-V3.1-Terminus的128K长上下文扩展数据完全对齐。

密集预热阶段。我们首先使用一个短暂的预热阶段来初始化闪电索引器。在此阶段，我们保持密集注意力机制，并冻结除闪电索引器外的所有模型参数。为使索引器输出与主注意力分布对齐，对于第{v*}个查询标记，我们首先通过汇总所有注意力头的注意力分数进行聚合。

[1]We illustrate the difference between the MQA and MHA modes of MLA in Appendix A.
[2]`https://huggingface.co/deepseek-ai/DeepSeek-V3.2-Exp/tree/main/inference`

This sum is then L1-normalized along the sequence dimension to produce a target distribution $p_{t,:} \in \mathbb{R}^t$. Based on $p_{t,:}$, we set a KL-divergence loss as the training objective of the indexer:

$$\mathcal{L}^I = \sum_t \mathbb{D}_{\mathrm{KL}}\big(p_{t,:} \,\big\|\, \mathrm{Softmax}(I_{t,:})\big). \tag{3}$$

For warm-up, we use a learning rate of 10^{-3}. We train the indexer for only 1000 steps, with each step consisting of 16 sequences of 128K tokens, resulting in a total of 2.1B tokens.

Sparse Training Stage. Following indexer warm-up, we introduce the fine-grained token selection mechanism and optimize all model parameters to adapt the model to the sparse pattern of DSA. In this stage, we also keep aligning the indexer outputs to the main attention distribution, but considering only the selected token set $\mathcal{S}_t = \{s \,|\, I_{t,s} \in \text{Top-k}(I_{t,:})\}$:

$$\mathcal{L}^I = \sum_t \mathbb{D}_{\mathrm{KL}}\big(p_{t,\mathcal{S}_t} \,\big\|\, \mathrm{Softmax}(I_{t,\mathcal{S}_t})\big). \tag{4}$$

It is worth noting that we detach the indexer input from the computational graph for separate optimization. The training signal of the indexer is from only $\mathcal{L}^I$, while the optimization of the main model is according to only the language modeling loss. In this sparse training stage, we use a learning rate of 7.3×10^{-6}, and select 2048 key-value tokens for each query token. We train both the main model and the indexer for 15000 steps, with each step consisting of 480 sequences of 128K tokens, resulting in a total of 943.7B tokens.

2.2. Post-Training

After continued pre-training, we perform post-training to create the final DeepSeek-V3.2-Exp. The post-training of DeepSeek-V3.2-Exp also employs sparse attention in the same way as the sparse continued pre-training stage. In pursuit of a rigorous assessment of the impact of introducing DSA, for DeepSeek-V3.2-Exp, we maintain the same post-training pipeline, algorithm, and data as used for DeepSeek-V3.1-Terminus, which are detailed as follows.

Specialist Distillation. For each task, we initially develop a specialized model dedicated exclusively to that particular domain, with all specialist models being fine-tuned from the same pre-trained DeepSeek-V3.2 base checkpoint. In addition to writing tasks and general question-answering, our framework encompasses five specialized domains: mathematics, competitive programming, general logical reasoning, agentic coding, and agentic search. Each specialist is trained with large-scale Reinforcement Learning (RL) computing. Furthermore, we employ different models to generate training data for long chain-of-thought reasoning (thinking mode) and direct response generation (non-thinking mode). Once the specialist models are prepared, they are used to produce the domain-specific data for the final checkpoint. Experimental results demonstrate that models trained on the distilled data achieve performance levels only marginally below those of domain-specific specialists, with the performance gap being effectively eliminated through subsequent RL training.

Mixed RL Training. For DeepSeek-V3.2-Exp, we still adopt Group Relative Policy Optimization (GRPO) (DeepSeek-AI, 2025; Shao et al., 2024) as the RL training algorithm. Unlike in previous DeepSeek models, which are trained with multi-stage reinforcement learning, we merge reasoning, agent, and human alignment training into one RL stage. This approach effectively balances performance across diverse domains while circumventing the catastrophic forgetting issues commonly associated with multi-stage training paradigms. For reasoning and

该和随后沿序列维度进行L1归一化，生成目标分布$p_{t,:} \in \mathbb{R}^t$。基于$p_{t,:}$，我们将KL散度损失设定为索引器的训练目标：

$$\mathcal{L}^I = \sum_t \mathbb{D}_{\mathrm{KL}}\left(p_{t,:} \,\middle\|\, \mathrm{Softmax}(I_{t,:})\right). \tag{3}$$

作为预热，我们采用10^{-3}的学习率。我们仅对索引器进行1000步训练，每一步包含16个序列，每个序列有128K个标记，总计达到21亿个标记。

稀疏训练阶段。在索引器预热之后，我们引入细粒度令牌选择机制并优化所有模型参数，使模型适应DSA的稀疏模式。在此阶段，我们继续将索引器输出与主注意力分布对齐，但仅考虑选定的令牌集$\mathcal{S}_t = \{s \mid I_{t,s} \in \text{Top-k}(I_{t,:})\}$：

$$\mathcal{L}^I = \sum_t \mathbb{D}_{\mathrm{KL}}\left(p_{t,\mathcal{S}_t} \,\middle\|\, \mathrm{Softmax}(I_{t,\mathcal{S}_t})\right). \tag{4}$$

值得注意的是，我们将索引器的输入从计算图中分离出来进行独立优化。索引器的训练信号仅来自$\mathcal{L}^I$，而主模型的优化则仅依据语言建模损失。在此稀疏训练阶段，我们使用7.3×10^{-6}的学习率，并为每个查询令牌选择2048个键值令牌。我们对主模型和索引器均进行15000步训练，每步包含480个128K令牌的序列，总计处理943.7B个令牌。

2.2. 训练后处理

在持续预训练之后，我们执行后训练以创建最终的DeepSeek-V3.2-Exp。DeepSeek-V3.2-Exp的后训练同样采用了稀疏注意力机制，其方式与稀疏持续预训练阶段相同。为了严格评估引入DSA的影响，对于DeepSeek-V3.2-Exp，我们保持了与DeepSeek-V3.1-Terminus相同的后训练流程、算法和数据，具体细节如下。

专家蒸馏。针对每项任务，我们首先开发专门针对该特定领域的专用模型，所有专家模型均基于同一预训练的DeepSeek-V3.2基础检查点进行微调。除写作任务和通用问答外，我们的框架涵盖五个专业领域：数学、竞赛编程、通用逻辑推理、智能体编码和智能体搜索。每个专家模型均通过大规模强化学习（RL）计算进行训练。此外，我们采用不同模型来生成长链思维推理（思考模式）和直接响应生成（非思考模式）的训练数据。专家模型准备就绪后，将用于生成最终检查点所需的领域特定数据。实验结果表明，基于蒸馏数据训练的模型性能仅略低于领域专家模型，且通过后续RL训练可有效消除性能差距。

混合强化学习训练。对于DeepSeek-V3.2-Exp，我们仍采用分组相对策略优化（GRPO）（DeepSeek-AI，2025；Shao等人，2024）作为强化学习训练算法。与先前采用多阶段强化学习的DeepSeek模型不同，我们将推理、智能体和人类对齐训练合并至单一强化学习阶段。该方法在有效平衡多领域性能的同时，规避了多阶段训练范式常见的灾难性遗忘问题。针对推理与

	Benchmark (Metric)	DeepSeek-V3.1-Terminus	DeepSeek-V3.2-Exp
General	MMLU-Pro (EM)	85.0	85.0
	GPQA-Diamond (Pass@1)	80.7	79.9
	Humanity's Last Exam (Pass@1)	21.7	19.8
Search Agent	BrowseComp (Acc.)	38.5	40.1
	BrowseComp_zh (Acc.)	45.0	47.9
	SimpleQA (Acc.)	96.8	97.1
Code	LiveCodeBench (2408-2505) (Pass@1)	74.9	74.1
	Codeforces-Div1 (Rating)	2046	2121
	Aider-Polyglot (Acc.)	76.1	74.5
Code Agent	SWE Verified (Agent mode)	68.4	67.8
	SWE-bench Multilingual (Agent mode)	57.8	57.9
	Terminal-bench (Terminus 1 framework)	36.7	37.7
Math	AIME 2025 (Pass@1)	88.4	89.3
	HMMT 2025 (Pass@1)	86.1	83.6

Table 1 | Evaluations of DeepSeek-V3.1-Terminus and DeepSeek-V3.2-Exp. Overall, DeepSeek-V3.2-Exp does not show substantial performance degradation compared with DeepSeek-V3.1-Terminus. The performance of DeepSeek-V3.2-Exp on GPQA, HLE, and HMMT 2025 is lower than that of DeepSeek-V3.1-Terminus because DeepSeek-V3.2-Exp generates fewer reasoning tokens. However, this performance gap closes when using intermediate checkpoints that produce a comparable number of tokens.

agent tasks, we employ rule-based outcome reward, length penalty, and language consistency reward. For general tasks, we employ a generative reward model where each prompt has its own rubrics for evaluation. Our reward design carefully balances two key trade-offs: (1) length versus accuracy and (2) language consistency versus accuracy.

3. Evaluations

Model Capabilities. We evaluate DeepSeek-V3.2-Exp on a suite of benchmarks, which focus on diverse capabilities, and compare it with DeepSeek-V3.1-Terminus in Table 1. While DeepSeek-V3.2-Exp significantly improves computational efficiency on long sequences, we do not observe substantial performance degradation compared with DeepSeek-V3.1-Terminus, on both short- and long-context tasks. In addition, we also compare the reinforcement learning training curves of DeepSeek-V3.2-Exp and DeepSeek-V3.1-Terminus, as shown in Figure 2. The performance of both models on BrowseComp and SWE Verified improves steadily throughout the training process, with closely aligned curves, which reflects the training stability of DSA.

Inference Costs. DSA reduces the core attention complexity of the main model from $O(L^2)$ to $O(Lk)$, where k ($\ll L$) is the number of selected tokens. Although the lightning indexer still has a complexity of $O(L^2)$, it requires much less computation compared with MLA in DeepSeek-V3.1-Terminus. Combined with our optimized implementation, DSA achieves a significant end-to-end speedup in long-context scenarios. Figure 3 presents how token costs of DeepSeek-V3.1-Terminus and DeepSeek-V3.2-Exp vary with the token position in the sequence. These costs are estimated from benchmarking the actual service deployed on H800 GPUs, at

	Benchmark (Metric)	DeepSeek-V3.1-Terminus	DeepSeek-V3.2-Exp
General	MMLU-Pro (EM)	85.0	85.0
	GPQA-Diamond (Pass@1)	80.7	79.9
	Humanity's Last Exam (Pass@1)	21.7	19.8
Search Agent	BrowseComp (Acc.)	38.5	40.1
	BrowseComp_zh (Acc.)	45.0	47.9
	SimpleQA (Acc.)	96.8	97.1
Code	LiveCodeBench (2408-2505) (Pass@1)	74.9	74.1
	Codeforces-Div1 (Rating)	2046	2121
	Aider-Polyglot (Acc.)	76.1	74.5
Code Agent	SWE Verified (Agent mode)	68.4	67.8
	SWE-bench Multilingual (Agent mode)	57.8	57.9
	Terminal-bench (Terminus 1 framework)	36.7	37.7
Math	AIME 2025 (Pass@1)	88.4	89.3
	HMMT 2025 (Pass@1)	86.1	83.6

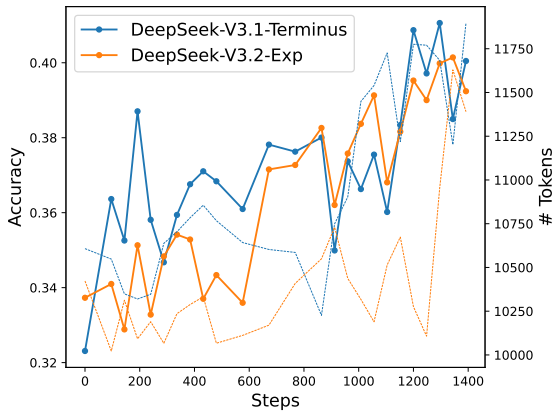
表1 | DeepSeek-V3.1-Terminus与DeepSeek-V3.2-Exp的评估结果。总体而言，DeepSeek-V3.2-Exp相比DeepSeek-V3.1-Terminus未出现明显的性能下降。DeepSeek-V3.2-Exp在GPQA、HLE和HMMT 2025上的表现略低于DeepSeek-V3.1-Terminus，这是因为DeepSeek-V3.2-Exp生成的推理标记数量较少。不过，当使用产生相当数量标记的中间检查点时，这一性能差距会消失。

在智能体任务中，我们采用基于规则的结果奖励、长度惩罚和语言一致性奖励。对于通用任务，我们采用生成式奖励模型，每个提示都有其独特的评估标准。我们的奖励设计精心平衡了两个关键权衡：（1）长度与准确性；（2）语言一致性与准确性。

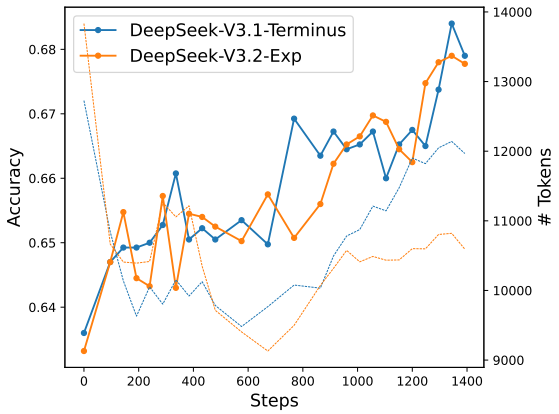
3. 评估

模型能力评估。我们在涵盖多种能力的基准测试套件上对DeepSeek-V3.2-Exp进行了评估，并在表1中将其与DeepSeek-V3.1-Terminus进行对比。虽然DeepSeek-V3.2-Exp在长序列计算效率上有显著提升，但在短上下文和长上下文任务中，我们并未观察到其性能相较DeepSeek-V3.1-Terminus出现明显下降。此外，我们还对比了如图2所示的两个模型的强化学习训练曲线。在训练过程中，两款模型在BrowseComp和SWE Verified任务上的表现均稳步提升，且训练曲线高度吻合，这体现了DSA训练框架的稳定性。

推理成本。DSA将主模型的核心注意力复杂度从$O(L^2)$降低至$O(Lk)$，其中k ($\ll L$)为所选令牌数量。尽管闪电索引器的复杂度仍为$O(L^2)$，但与DeepSeek-V3.1-Terminus中的MLA相比，其计算量大幅减少。结合我们的优化实现，DSA在长上下文场景中实现了显著的端到端加速。图3展示了DeepSeek-V3.1-Terminus和DeepSeek-V3.2-Exp的令牌成本如何随序列中令牌位置变化。这些成本基于部署在H800 GPU上的实际服务基准测试得出。

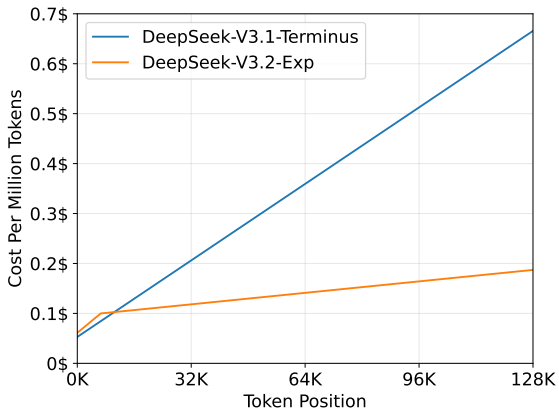


(a) BrowseComp Training Curve

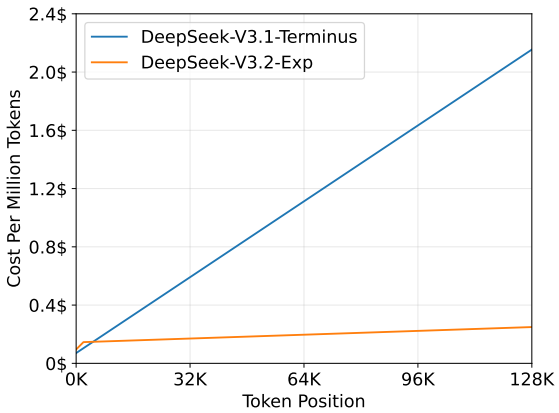


(b) SWE Training Curve

Figure 2 | RL training curve of DeepSeek-V3.1-Terminus and DeepSeek-V3.2-Exp on BrowseComp and SWE Verified. The solid and dashed lines denote the accuracy and average output tokens, respectively.



(a) Prefilling



(b) Decoding

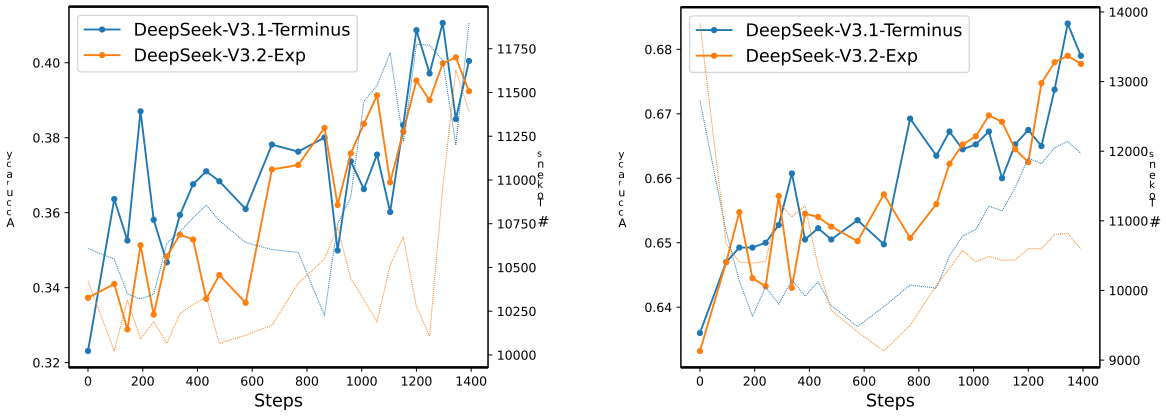
Figure 3 | Inference costs of DeepSeek-V3.1-Terminus and DeepSeek-V3.2-Exp on H800 clusters.

a rental price of 2 USD per GPU hour. Note that for short-sequence prefilling, we specially implement a masked MHA mode to simulate DSA, which can achieve higher efficiency under short-context conditions.

Future Validation in Real World. Although our internal evaluations show promising results of DeepSeek-V3.2-Exp, we are actively pursuing further large-scale testing in real-world scenarios to uncover potential limitations of the sparse attention architecture.

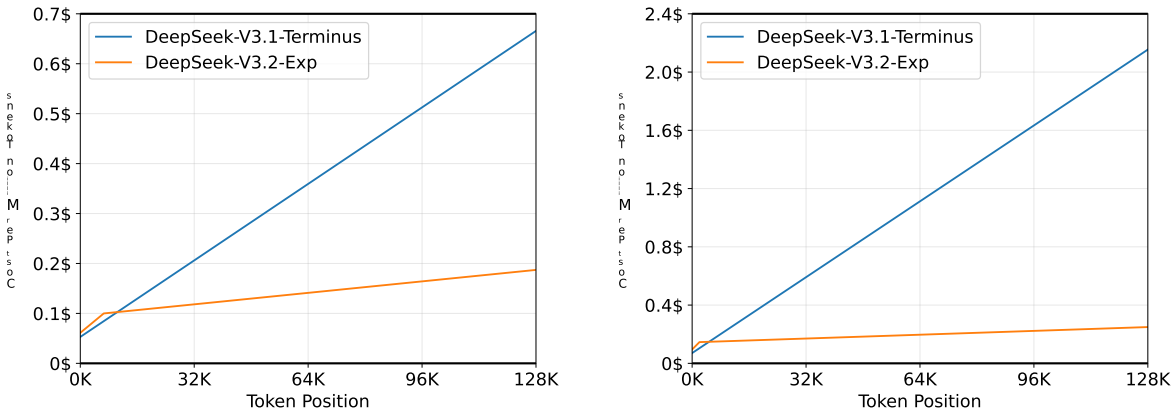
References

DeepSeek-AI. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. CoRR, abs/2405.04434, 2024. doi: 10.48550/ARXIV.2405.04434. URL https://doi.org/10.48550/arXiv.2405.04434.



(a) BrowseComp Training Curve (b) SWE Training Curve

图2 | DeepSeek-V3.1-Terminus和DeepSeek-V3.2-Exp在BrowseC-omp和SWE Verified上的RL训练曲线。实线和虚线分别表示准确率和平均输出标记数。



(a) Prefilling (b) Decoding

图3 | DeepSeek-V3.1-Terminus与DeepSeek-V3.2-Exp在H800集群上的推理成本。

租赁价格为每小时2美元每GPU。请注意，针对短序列预填充，我们专门实现了掩码多头注意力模式来模拟DSA，该模式可在短上下文条件下实现更高效率。

未来在真实世界中的验证。虽然我们的内部评估显示DeepSeek-V3.2-Exp取得了令人鼓舞的结果，但我们仍在积极推动在真实场景中进行大规模测试，以揭示稀疏注意力架构的潜在局限性。

参考文献

DeepSeek-AI. Deepseek-v2：一个强大、经济且高效的专家混合语言模型。CoRR, abs/2405.04434, 2024. doi: 10.48550/ARXIV.2405.04434. URL `https://doi.org/10.48550/arXiv.2405.04434`.

DeepSeek-AI. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. Nature, 645(8081):633–638, 2025.

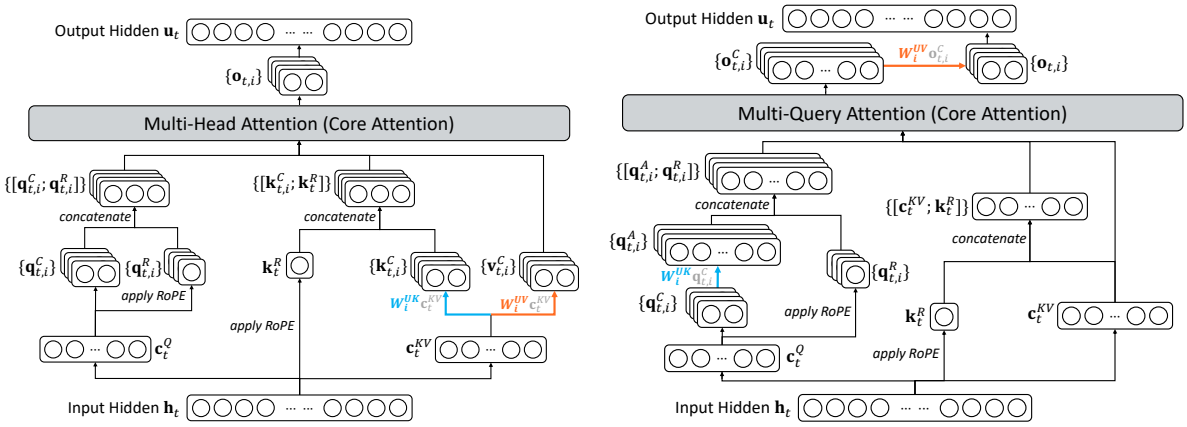
Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, M. Zhang, Y. K. Li, Y. Wu, and D. Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. CoRR, abs/2402.03300, 2024. doi: 10.48550/ARXIV.2402.03300. URL https://doi.org/10.48550/arXiv.2402.03300.

N. Shazeer. Fast transformer decoding: One write-head is all you need. CoRR, abs/1911.02150, 2019. URL http://arxiv.org/abs/1911.02150.

J. Yuan, H. Gao, D. Dai, J. Luo, L. Zhao, Z. Zhang, Z. Xie, Y. Wei, L. Wang, Z. Xiao, Y. Wang, C. Ruan, M. Zhang, W. Liang, and W. Zeng. Native sparse attention: Hardware-aligned and natively trainable sparse attention. In W. Che, J. Nabende, E. Shutova, and M. T. Pilehvar, editors, Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, pages 23078–23097. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.acl-long.1126/.

Appendices

A. MHA and MQA Modes of MLA



(a) MHA mode of MLA.

(b) MQA mode of MLA.

Figure 4 | Illustration of the MHA and MQA modes of MLA. For DeepSeek-V3.1-Terminus, the MHA mode is used for training and prefilling, while the MQA mode is used for decoding.

Figure 4 illustrates two aspects of MLA – the MHA and MQA modes – as well as the transformation between them.

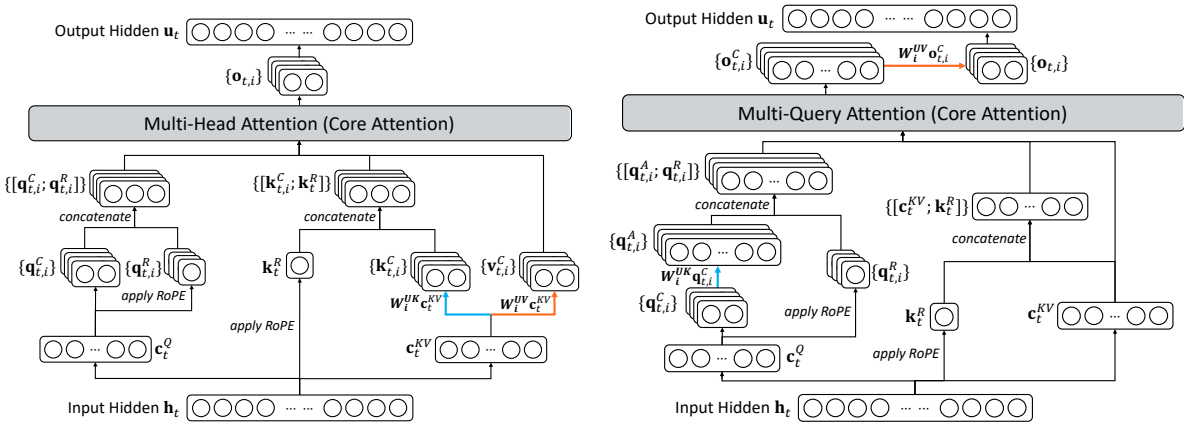
DeepSeek-AI。Deepseek-r1 通过强化学习激励大型语言模型进行推理。《自然》杂志，645(8081)：633–638，2025年。

邵哲、王鹏、朱强、徐荣、宋杰、张明、李永坤、吴岩与郭栋。DeepSeek-Math：突破开放语言模型数学推理的极限。《CoRR》期刊，论文编号 abs/2402.03300，2024年。DOI：10.48550/ARXIV.2402.03300。网址 https://doi.org/10.48550/arXiv.2402.03300。

N. Shazeer. 快速Transformer解码：一个写头足矣。CoRR, abs/1911.02150, 2019. URL http://arxiv.org/abs/1911.02150。 袁健、高航、戴德建、罗金、赵磊、张钊、谢哲、魏勇、王磊、肖哲、王宇、阮超、张明、梁伟、曾文。原生稀疏注意力：硬件对齐且可原生训练的稀疏注意力。见：车伟、Nabende J.、Shutova E.、Pilehvar M.T.编，计算语言学协会第63届年会论文集（第一卷：长论文），ACL 2025，第23078–23097页。计算语言学协会，2025。URL https://aclanthology.org/2025.acl-long.1126/。

附录

A. MLA的多头注意力（MHA）与多查询注意力（MQA）模式



(a) MHA mode of MLA. (b) MQA mode of MLA.

图4 | MLA的多头注意力（MHA）与多查询注意力（MQA）模式示意图。对于DeepSeek-V3.1-Terminus模型，训练和前填充阶段采用MHA模式，而解码阶段则使用MQA模式。

图4展示了MLA的两个方面——MHA与MQA模式——以及它们之间的转换关系。