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